

Циклова комісія Комп'ютерна інженерія

Методичні рекомендації до виконання лабораторних робіт з дисципліни «Технології (Програмування)»

Розробив: _____ О.В.Старосельцева
Розглянуто та схвалено
на засіданні предметної (циклової)
комісії комп'ютерної інженерії
Протокол № 1 від 01.09.2022 р.
Голова комісії _____ О.В. Старосельцева

Зміст

- 1 Лабораторна робота № 1.
Складання та налагодження лінійних програм.
- 2 Лабораторна робота № 2.
Складання та налагодження розгалужених програм з використанням операторів if та if-else.
- 3 Лабораторна робота № 3.
Складання та налагодження розгалужених програм з використанням вкладених операторів if та if-else.
- 4 Лабораторна робота № 4.
Складання та налагодження розгалужених програм з використанням оператора-перемикача switch.
- 5 Лабораторна робота № 5.
Складання та налагодження програм з використанням ітераційних циклів.
- 6 Лабораторна робота № 6.
Складання та налагодження програм з використанням регулярних циклів.
- 7 Лабораторна робота № 7.
Складання та налагодження програм обробки одномірних масивів
- 8 Лабораторна робота № 8.
Складання та налагодження програм пошуку екстремумів лінійних масивів.
- 9 Лабораторна робота № 9.
Складання та налагодження програм сортування лінійних масивів.
- 10 Лабораторна робота № 10.
Складання та налагодження програм обробки матриць
- 11 Лабораторна робота № 11.
Складання та налагодження програм з використанням масивів структур
- 12 Лабораторна робота № 12
Складання та налагодження програм обробки динамічних матриць.
- 13 Лабораторна робота № 13.
Складання та налагодження програм з використанням функцій
- 14 Лабораторна робота № 14.
Складання та налагодження програм з використанням функцій, які приймають параметри-масиви.
- 15 Лабораторна робота № 15.
Складання та налагодження програм обробки рекурсивних математичних функцій.
- 16 Лабораторна робота № 16
Складання та налагодження програм роботи з текстовими файлами.
- 17 Лабораторна робота № 17.
Складання програм обробки бінарних файлів в стилі C++.

Методичні рекомендації
до виконання лабораторної роботи № 1
з дисципліни Технології (Програмування)

Тема: Складання та налагодження лінійних програм.

Мета: Придбати практичні навички при написанні та налагодженні програм лінійних алгоритмів на мові програмування C++.

ЧАС 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Оголошення змінних

- Кожна змінна програми повинна бути оголошена;
- Оголошення змінних звичайно поміщають на початку функції, відразу за заголовком. Варто звернути увагу, що хоча мова c++ допускає оголошення змінних практично в будь-якому місці функції, повідомляти змінні краще все-таки на початку функції, доповнюючи інструкцію оголошення коротким коментарем про призначення змінної;
- Інструкція оголошення змінної виглядає так:
<Тип> <Ім'я змінної > ;

Наприклад:

```
int x;  
float f, Z1;
```

- Інструкцію оголошення змінної можна використати для ініціалізації змінної. У цьому випадку оголошення змінної записують у такий спосіб:

<Тип> <Ім'я змінної > = <Початкове значення>;

Наприклад:

```
float a; //Кут альфа
```

- В імені змінної можна використати букви латинського алфавіту й цифри (першим символом повинна бути буква);
- Компілятор c++ розрізняє прописні й малі літери, тому, наприклад, імена `summa` й `summa` позначають різні змінні;
- Основними числовими типами мови c++ є: `int` (цілий) і `float` (дробовий);
- Після інструкції оголошення змінної рекомендується за допомогою коментаря вказувати призначення змінної. Наприклад:

```
int x=-7;  
float f,a=15.27;
```

2 Інструкція присвоювання

- Інструкція присвоювання призначена для зміни значень змінних, у тому числі й для обчислень "по формулі";
- У відмінність більшості мов програмування, у c++ інструкція присвоювання, що виконує деяку дію, може бути записана декількома способами, наприклад, замість $x=x+dx$ можна записати $x+=dx$, а замість $i=i+1$ скористатися оператором інкремента й записати $i++$;
- Значення вираження в лівій частині інструкції присвоювання залежить від типу операндів й операції, що виконуються над операндами. Цілочисельні додавання та віднімання виконуються без обліку переповнення. Наприклад, якщо змінна x , що оголошена як `int`, має значення 32767, то в результаті виконання інструкції $x=x+1$ значення змінної x буде дорівнює -32768;
- Результатом виконання операції ділення над цілими операндами є ціле, що отримується відкиданням дробової частини результату розподілу;
- Приклад інструкції присвоювання змінній $z1$ значення, отриманого за формулою: $Z_1 = -4 \sin^2\left(\frac{\alpha - \beta}{2}\right) \cdot \cos(\alpha + \beta)$

```
z1=-4*sin((a-b)/2)*sin((a-b)/2)*cos(a+b);  
// змінна a - кут  $\alpha$ , змінна b - кут  $\beta$ 
```

3 Введення даних

- Для введення даних з клавіатури в стилі Cі використовуються функції, які описуються в бібліотечному файлі `stdio.h`. Це, насамперед, функція `scanf()`;
- Першим параметром функції `scanf()` є керуючий рядок, інші параметри - адреси змінних, значення яких повинні бути уведені;
- Керуючий рядок являє собою укладений у подвійні лапки список специфікаторів: `%i` - для введення цілих чисел зі знаком, `%u` - для введення беззнакових цілих, `%f` - для введення дійсних чисел, `%c` - для введення символу, `%s` - для введення рядка символів;

– Використання імені змінної, а не її адреси, як параметр функції `scanf()` є типовою помилкою починаючих програмістів. До речі, компілятор цю помилку не виявляє.

– Приклади використання функції `scanf()`:

```
int m;
. . .
scanf("%i",&m); /* Ввести з клавіатури ціле число
                і присвоїти його змінній m */

float x, y;
. . .
scanf("%f%f",&x,&y); /* Ввести з клавіатури два дійсних
                числа і присвоїти їх відповідним змінним x та y */
```

– При використанні бібліотеки класів C++, використовується бібліотечний файл `iostream.h`, у якому визначені стандартний потік введення `cin`, а також відповідна операції: `>>` – операція читання даних з потоку.

– Загальна форма запису читання даних з потоку введення:

– `Cin>>змінна1[>>змінна2...>>зміннаn];`

– В якості змінних вказуються імена об'єктів, куди будуть записані дані, прочитані з потоку введення. Наприклад:

```
int m;
. . .
cin>>m; /* Ввести з клавіатури число
        і присвоїти його змінній m */

float x, y;
. . .
cin>>x>>y; /* Ввести з клавіатури два числа
            і присвоїти їх відповідним змінним x та y */
```

4 Виведення

- Для виведення на екран монітора повідомлень та значень змінних в стилі Cі використовуються функції, які описуються в бібліотечному файлі `stdio.h`. Це, насамперед, функція `printf()` ;

- Першим параметром функції `printf()` є рядок виведення, який задає текст, що виводиться та формат виведення значень змінних, імена яких зазначені в якості інших параметрів функції;

- Формат виведення значень змінних задається за допомогою специфікатора перетворення - послідовності символів, що починається із символу `%` ;

- При виведенні числових значень найбільше часто використовуються наступні специфікатори: `%i` - для виведення цілих зі знаком, `%u` - для виведення беззнакових цілих, `%f` - для виведення дійсних значень, у вигляді числа із плаваючою крапкою, `%n, mf` - для у форматі з фіксованою крапкою, де `n` - кількість цифр цілої частини, `m` - дробової;

- Деякі символи можуть бути поміщені в рядок виведення тільки як послідовність інших, звичайних символів: `\n` - новий рядок, `\t` - табуляція, `\"` - подвійні лапки, `\\` - символ `\`;

- Приклад використання функції `printf()` для виведення значення змінної `Z1`:

```
printf("\n Значение Z1 равно  \t%10.5f ",Z1);
```

- Поряд з функцією `printf()`, для виведення на екран повідомлень можна використати функцію `puts`, що після виведення тексту автоматично переводить курсор у початок наступного рядка;

- Щоб відразу після закінчення роботи програми вікно, у якому програма працювала, не було автоматично перекрите іншим вікном, наприклад вікном редактору тексту середовища розробки або панелями `norton commander`, у кінець програми потрібно вставити наступні дві інструкції:

```
printf("Для завершення натисніть клавішу <Enter>");  
getch();
```

- При використанні бібліотеки класів C++, використовується бібліотечний файл `iostream.h`, у якому визначені стандартний потік

виведення даних на екран монітора `cout`, а також відповідна операція:

>> – операція читання даних з потоку.

– Загальна форма запису даних в потік виведення:

```
cout<<"Рядок_даних_1" [<<"Рядок_даних_2"...<<"Рядок_даних_n"];
```

– Рядок даних може містити:

– символи, які безпосередньо будуть виводитися на екран (проста текстова інформація);

– керуючі послідовності символів.

– Аналогічно, як при використанні керуючих символів у функції виведення `printf()`, вони не виводяться на екран, а керують розташуванням символів, які виводяться на екран монітора. Ознакою керуючого символу є наявність символу `"\"`. У рядок формату можуть входити наступні основні керуючі послідовності символів:

– `'\n'` – новий рядок;

– `'\t'` – горизонтальна табуляція;

– `'\v'` – вертикальна табуляція;

– `'\b'` – повернення на символ;

– `'\r'` – повернення на початок рядка;

– `\a` – звуковий сигнал.

– Зазвичай, для переведення на новий рядок використовують вираз `endl`.

– Приклади використання потоку виведення для виведення на екран монітору значення змінних:

```
int a=5;
```

```
cout<<"Значення змінної a"<<a;
```

Результат роботи програми:

Значення змінної a=5.

```
float x=2.78;
```

```
cout<<"Значення змінної x"<<x;
```

Результат роботи програми:

Значення змінної x=2.78.

5 Програми з лінійною структурою

- Програми з лінійною структурою є найпростішими й використовуються, як правило, для реалізації простих обчислень по формулах;
- Програмах з лінійною структурою інструкції виконуються послідовно, одна за іншою;
- Алгоритм програми з лінійною структурою може бути представлений у вигляді блок-схеми, показаної на малюнку.



ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Для заданих користувачем вхідних даних скласти та налагодити програму обчислення значення Z_1 та Z_2 за формулами згідно Вашого варіанту.

Програма повинна задовольняти наступним вимогам:

- для введення значення вхідних даних використовувати функцію `scanf()`;
- для виведення значення Z_1 використовувати функцію `printf()`;
- для виведення значення Z_2 використовувати потік виведення `cout`;
- введення та виведення даних повинно містити необхідні для користувача повідомлення, для виведення яких використовувати потік виведення `cout`.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Вхідні дані	Формули для обчислення
1	Кут α – значення вводиться в радіанах	$Z_1 = 2 \sin^2(3\pi - 2\alpha) \cdot \cos^2(5\pi + 2\alpha)$ $Z_2 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5\pi \cdot 8\alpha}{2}\right)$
2	Кут α – значення вводиться в радіанах	$Z_1 = \cos \alpha + \sin \alpha + \cos^3 \alpha + \sin 3\alpha$ $Z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right)$
3	Кут α – значення вводиться в радіанах	$Z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2 \sin^2 2\alpha}$ $Z_2 = 2 \sin \alpha$
4	Кут α – значення вводиться в радіанах	$Z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha - \cos 3\alpha + \cos 5\alpha}$ $Z_2 = \operatorname{tg} 3\alpha$

Варіант	Вхідні дані	Формули для обчислення
5	Кут α – значення вводиться в радіанах	$Z_1 = 1 - \frac{1}{4} \sin^2 2\alpha + \cos 2\alpha$ $Z_2 = \cos^2 \alpha + \cos^4 \alpha$
6	Кут α – значення вводиться в радіанах	$Z_1 = \cos \alpha + \cos 2\alpha + \cos 6\alpha + \cos 7\alpha$ $Z_2 = 4 \cos \frac{\alpha}{2} \cdot \cos \frac{5\alpha}{2} \cdot \cos 4\alpha$
7	Кут α – значення вводиться в радіанах	$Z_1 = \cos^2 \left(\frac{3\pi}{8} - \frac{\alpha}{4} \right) - \cos^2 \left(\frac{11\pi}{8} + \frac{\alpha}{4} \right)$ $Z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}$
8	Кути x та y – значення вводиться в радіанах	$Z_1 = \cos^2 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1$ $Z_2 = \sin(x + y) \cdot \sin(x - y)$
9	Кути α та β – значення вводиться в радіанах	$Z_1 = (\cos \alpha - \cos \beta)^2 - (\sin \alpha + \sin \beta)^2$ $Z_2 = -4 \sin^2 \left(\frac{\alpha - \beta}{2} \right) \cdot \cos(\alpha + \beta)$
10	Кут α – значення вводиться в радіанах	$Z_1 = \frac{\sin \left(\frac{\pi}{2} + 3\alpha \right)}{1 - \sin(3\alpha - \pi)}$ $Z_2 = \operatorname{ctg} \left(\frac{5}{4} \pi + \frac{3}{2} \alpha \right)$
11	Кут α – значення вводиться в радіанах	$Z_1 = \frac{1 - 2 \sin^2 \alpha}{1 + \sin 2\alpha}$ $Z_2 = \frac{1 - \operatorname{tg} \alpha}{1 + \operatorname{tg} \alpha}$
12	Кут α – значення вводиться в радіанах	$Z_1 = \frac{\sin 4\alpha}{1 + \cos 4\alpha} \cdot \frac{\cos 2\alpha}{1 + \cos 2\alpha}$ $Z_2 = \operatorname{ctg} \left(\frac{3}{2} \alpha - \alpha \right)$
13	Кути α та β – значення вводиться в радіанах	$Z_1 = \frac{\sin \alpha + \cos(2\beta - \alpha)}{\cos \alpha - \sin(2\beta - \alpha)}$ $Z_2 = \frac{1 + \sin 2\beta}{\cos 2\beta}$
14	Кут α – значення вводиться в радіанах	$Z_1 = \frac{\cos \alpha + \sin \alpha}{\cos \alpha - \sin \alpha}$ $Z_2 = \operatorname{tg} 2\alpha + \sin 2\alpha$
15	Дійсне число b	$Z_1 = \frac{\sqrt{2b + 2\sqrt{b^2 - 4}}}{\sqrt{b^2 - 4} + b + 2}$ $Z_2 = \frac{1}{\sqrt{b + 2}}$

Варіант	Вхідні дані	Формули для обчислення
16	Дійсне число x	$Z_1 = \frac{x^2 + 2x - 3 + (x+1)\sqrt{x^2 - 9}}{x^2 - 2x - 3 + (x-1)\sqrt{x^2 - 9}}$ $Z_2 = \sqrt{\frac{x+3}{x-3}}$
17	Дійсне число m	$Z_1 = \frac{\sqrt{(3m+2)^2 - 24m}}{3\sqrt{m} - \frac{2}{\sqrt{m}}}$ $Z_2 = -\sqrt{m}$
18	Дійсне число a	$Z_1 = \left(\frac{a+2}{\sqrt{2a}} - \frac{a}{\sqrt{2a}+2} + \frac{2}{a-\sqrt{2a}} \right) \cdot \frac{\sqrt{a}-\sqrt{2}}{a+2}$ $Z_2 = \frac{1}{\sqrt{a}+\sqrt{2}}$
19	Дійсне число a	$Z_1 = \left(\frac{1+a+a^2}{2a+a^2} + 2 - \frac{1-a+a^2}{2a-a^2} - 1 \right) \cdot (5-4a)$ $Z_2 = \frac{4-a^2}{2}$
20	Дійсні числа m та n	$Z_1 = \frac{(m-1)\sqrt{m} - (n-1)\sqrt{n}}{\sqrt{m^3n} + nm + m^2 - m}$ $Z_2 = \frac{\sqrt{m} - \sqrt{n}}{m}$
21	Кут α – значення вводиться в радіанах	$Z_1 = \frac{\sin 2\alpha + \sin 5\alpha - \sin 3\alpha}{\cos \alpha + 1 - 2\sin^2 2\alpha}$ $Z_2 = 2\sin^2(3\pi - 2\alpha) \cdot \cos^2(5\pi + 2\alpha)$
22	Кут α – значення вводиться в радіанах	$Z_1 = \frac{1}{4} - \frac{1}{4} \sin\left(\frac{5\pi \cdot 8\alpha}{2}\right)$ $Z_2 = \operatorname{tg} 3\alpha$
23	Кут α – значення вводиться в радіанах	$Z_1 = \frac{1 - 2\sin^2 \alpha}{1 + \sin 2\alpha}$ $Z_2 = 4\cos \frac{\alpha}{2} \cdot \cos \frac{5\alpha}{2} \cdot \cos 4\alpha$
24	Кути α , x та y – значення вводиться в радіанах	$Z_1 = \cos^2 x + \sin^2 y + \frac{1}{4} \sin^2 2x - 1$ $Z_2 = 2\sqrt{2} \cos \alpha \cdot \sin\left(\frac{\pi}{4} + 2\alpha\right)$
25	Кут α – значення вводиться в радіанах	$Z_1 = \cos \alpha + \sin \alpha + \cos^3 \alpha + \sin 3\alpha$ $Z_2 = \cos^2 \alpha + \cos^4 \alpha$
26	Кути α та β – значення вводиться в радіанах	$Z_1 = -4\sin^2\left(\frac{\alpha - \beta}{2}\right) \cdot \cos(\alpha + \beta)$ $Z_2 = \frac{\sqrt{2}}{2} \sin \frac{\alpha}{2}$

КОНТРОЛЬНІ ПИТАННЯ

- 1 Які типи даних віднесені до простих?
- 2 Яку довжину має тип даних `int`?
- 3 Як визначаються модифікатори простих типів даних?
- 4 Яку довжину має дійсний тип даних?
- 5 Яку довжину має тип даних `double`?
- 6 Дайте визначення поняття змінна.
- 7 Чим відрізняється оголошення змінної від її визначення?
- 8 Що означає ключове слово `const`, яке передує опису змінної?
- 9 Яка функція використовується в оригінальному Сі для виведення даних на екран монітора?
- 10 Що таке "рядок формату" у функції виведення даних `printf()`?
- 11 Як називається потік в С++ для виведення даних на екран монітора?
- 12 Яка функція використовується в оригінальному Сі для введення даних з клавіатури?
- 13 Що таке "рядок формату" у функції введення даних `scanf()`?
- 14 Як називається потік в С++ для введення даних з клавіатури?
- 15 Які дані може містити "рядок формату" функції виведення даних `printf()`?
- 16 Що таке специфікатор перетворення (специфікатор формату)?
- 17 Назвіть специфікатор перетворення для експоненціального типу даних.
- 18 Яка операція використовується для запису даних у потік виведення даних в С++?
- 19 Які дані може містити "рядок формату" функції введення даних `scanf()`?
- 20 Назвіть специфікатор перетворення для цілого типу даних.
- 21 Який оператор використовується для взяття адреси змінної у функції `scanf()`?
- 22 Що таке "модифікатор формату" у функції виведення даних `printf()`?
Приведіть приклад.
- 23 Назвіть специфікатор перетворення для дійсного типу даних.
- 24 Яка операція використовується для читання даних з потоку введення даних в С++?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] /
ІО.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.

- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання)
[Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни
«Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 2
з дисципліни Технології (Програмування)

Тема: Складання та налагодження розгалужених програм з використанням операторів if та if-else.

Мета: Придбати практичні навички при складанні та налагодженні програм розгалужених алгоритмів, використовуючи оператори if та if-else.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Оператор розгалуження if - else

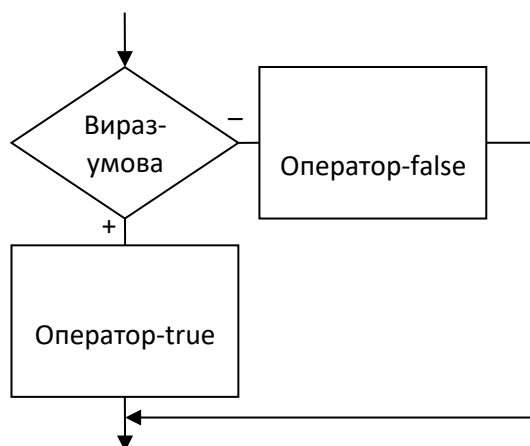
Загальні зауваження

- оператор if - else використовується для вибору одного із двох напрямків подальшого ходу програми;

- загальна (*повна*) форма запису оператора розгалуження наступна:

```
if (вираз - умова)      // ключового слова then в C++ немає  
    оператор_true;  
else                    // зверніть увагу на крапку з комою перед else  
    оператор_false;
```

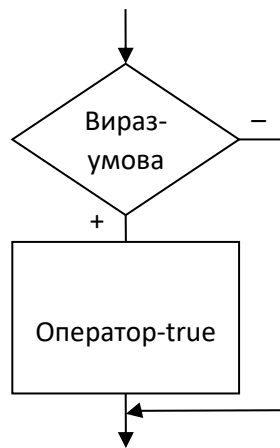
Їй відповідає наступна конструкція блок – схеми:



- вибір послідовності інструкцій здійснюється залежно від значення умови - ув'язненого в дужки виразу, записаного після if;

- інструкція, записана після else, виконується в тому випадку, якщо значення виразу – умови дорівнює нулю, у всіх інших випадках виконується інструкція, що впливає за умовою;
- якщо при дотриманні або недотриманні умови треба виконати кілька інструкцій програми, то ці інструкції варто об'єднати у групу - взяти у фігурні дужки { };
- оператор розгалуження може мати *скорочену* форму:

```
if (вираз-умова)
оператор-true;
```



В цьому випадку, якщо вираз – умова отримує значення 0, програма переходить до виконання наступного оператора;

Логічні операції порівняння

Операції порівняння (відношення) наведені у таблиці 1.

Результатом усякої операції відношення є числове значення типу int, рі-вне одиниці, якщо порівняння істинно, і нулю в протилежному випадку. Таким чином, операції відношення у внутрішньому машинному представленні при-водять до арифметичного результату.

Зауваження. Строго говорячи, логічне значення "істина" повинне від-повідати будь-якому числовому значенню, відмінному від нуля. Надалі ми побачимо, що саме така угода прийнята в мові С. Це дає можливість об'єдна-ти поняття арифметичного, умовного й логічного виразів у єдиному понятті "вираз", що дуже важливо з погляду гнучкості й "симетричності" мови. Вирази, сконструйовані за допомогою операцій відношення, прийнято називати умовними виразами. У процесі обчислення значення всякого умовного вира-зу, операції відношення обробляються зліва направо. Встановлений порядок може бути змінений шляхом внесення частини виразу в круглі дужки. Напри-клад, запис

$x < y == z$

повністю еквівалентна запису

$(x < y) == z$

у той час як вираз виду

$x < (y == z)$

відрізняється від попереднього порядком виконання операцій $<$ й $==$.

Результатом цих операцій є значення, відмінне від нуля, якщо операція порівняння істинна, або нуль, якщо операція порівняння хибна.

Таблиця 1. – Знаки операцій відношення

Знак операції	Призначення
$<$	Дорівнює true, якщо лівий операнд менше правого операнду
$>$	Дорівнює true, якщо лівий операнд більше правого операнду
$<=$	Дорівнює true, якщо лівий операнд менше або дорівнює правому операнду
$>=$	Дорівнює true, якщо лівий операнд більше або дорівнює правому операнду
$==$	Дорівнює true, якщо лівий операнд дорівнює правому операнду
$!=$	Дорівнює true, якщо лівий операнд не дорівнює правому операнду

В C++ існує три логічні операції:

Логічна операція І $\&\&$;

Логічна операція АБО $\|\;$;

Логічна операція НЕ $!$ або логічне заперечення.

Логічні операції утворюють складну (складену) умову з декількох простих (двох або більше) умов. Ці операції спрощують структуру програмного коду в декілька разів. Так, можна обійтися і без них, але тоді кількість іфів збільшується в кілька разів, залежно від умови. У наступній таблиці коротко охарактеризовано всі логічні операції в мові програмування C, для побудови логічних умов.

Таблиця 2 – Логічні операції C

Операції	Позначення	Умова	Короткий опис
І	&&	$a == 3 \ \&\& \ b > 4$	Складена умова істинна, якщо істинні обидві прості умови
АБО		$a == 3 \ \ b > 4$	Складена умова істинна, якщо істинна, хоча б одна з простих умов
НЕ	!	$!(a == 3)$	Умова істинна, якщо a не дорівнює 3

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Для заданого користувачем дійсного числа x скласти та налагодити програму обчислення значення функції згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- враховувати область визначення функції. При введенні вхідних даних, що не входять в область визначення функції, повинно бути виведено відповідне повідомлення

(наприклад: На введенних входних даних функция не определена)

- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення, для виведення яких використовувати потік виведення `cout`.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Формули для обчислення
1	$y = \frac{\sqrt{x-5}}{x-10}$
2	$y = \frac{\sqrt{x}-6}{x-5}$
3	$y = \frac{10-\sqrt{x}}{7-x}$
4	$y = \frac{1}{x-7} + \sqrt{x-3}$
5	$y = \frac{x-7}{\sqrt{x}-10}$
6	$y = \frac{1}{\sqrt{7-x}} + \frac{1}{10-x}$
7	$y = \frac{\sqrt{x+5}}{x-1}$
8	$y = \frac{x-5}{8-\sqrt{x}}$
9	$y = \sqrt{x+3} - \frac{x}{1-x}$
10	$y = \frac{x+6}{5-\sqrt{x}}$

Варіант	Формули для обчислення
11	$y = \frac{1}{\sqrt{7+x}} + \frac{1}{10-x}$
12	$y = \frac{x-6}{\sqrt{x}-4}$
13	$y = \frac{\sqrt{x}+3}{x-1}$
14	$y = \frac{\sqrt{x+3}}{x+2}$
15	$y = \frac{\sqrt{x-2}}{1} + \frac{1}{x-4}$
16	$y = \sqrt{x-3} - \frac{x}{5-x}$
17	$y = \frac{x-5}{\sqrt{x}-7}$
18	$y = \sqrt{6-x} - \frac{x}{10+x}$
19	$y = \frac{1}{\sqrt{3-x}} + \frac{1}{5-x}$
20	$y = \frac{\sqrt{x+2}}{1} + \frac{1}{1-x}$
21	$y = \frac{x+2}{\sqrt{x}-2}$

Варіант	Формули для обчислення
22	$y = \frac{\sqrt{x+5}}{\sqrt{x}-7}$
23	$y = \sqrt{1-x} - \frac{x}{2+x}$

КОНТРОЛЬНІ ПИТАННЯ

- 1 Наведіть повну форму коду умовного оператора if.
- 2 Виконайте загальний вигляд блок-схеми повного умовного оператора.
- 3 Наведіть скорочену форму коду умовного оператора if.
- 4 Виконайте загальний вигляд блок-схеми скороченого умовного оператора.
- 5 Що може включати "вираз-умова" в операторі if?
- 6 Що необхідно задіювати, якщо необхідне виконання не одного оператора, а деякого набору операторів в операторі if?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 3
з дисципліни Технології (Програмування)

Тема: Складання та налагодження розгалужених програм з використанням операторів if та if-else.

Мета: Придбати практичні навички при складанні та налагодженні програм розгалужених алгоритмів, використовуючи оператори if та if-else.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

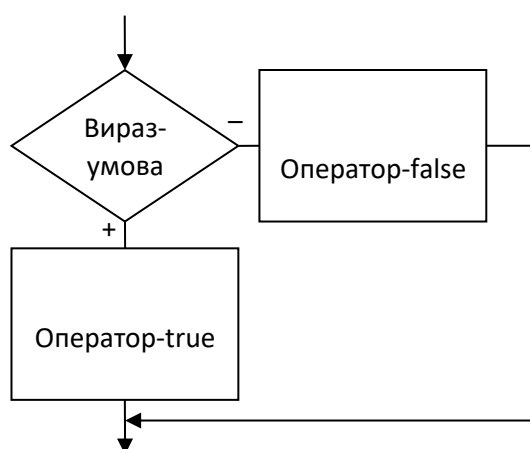
Оператор розгалуження if - else

Загальні зауваження

- оператор if - else використовується для вибору одного із двох напрямків подальшого ходу програми;
- загальна (*повна*) форма запису оператора розгалуження наступна:

```
if (вираз - умова)    // ключового слова then в C++ немає  
    оператор_true;  
else                  // зверніть увагу на крапку з комою перед else  
    оператор_false;
```

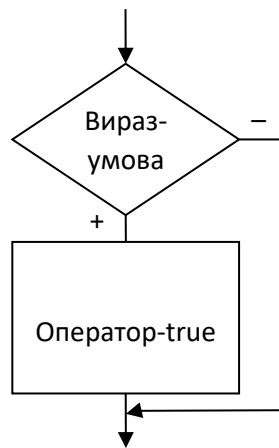
Їй відповідає наступна конструкція блок – схеми:



- вибір послідовності інструкцій здійснюється залежно від значення умови - ув'язненого в дужки виразу, записаного після if;

- інструкція, записана після else, виконується в тому випадку, якщо значення виразу – умови дорівнює нулю, у всіх інших випадках виконується інструкція, що впливає за умовою;
- якщо при дотриманні або недотриманні умови треба виконати кілька інструкцій програми, то ці інструкції варто об'єднати у групу - взяти у фігурні дужки { };
- оператор розгалуження може мати *скорочену* форму:

```
if (вираз-умова)
оператор-true;
```



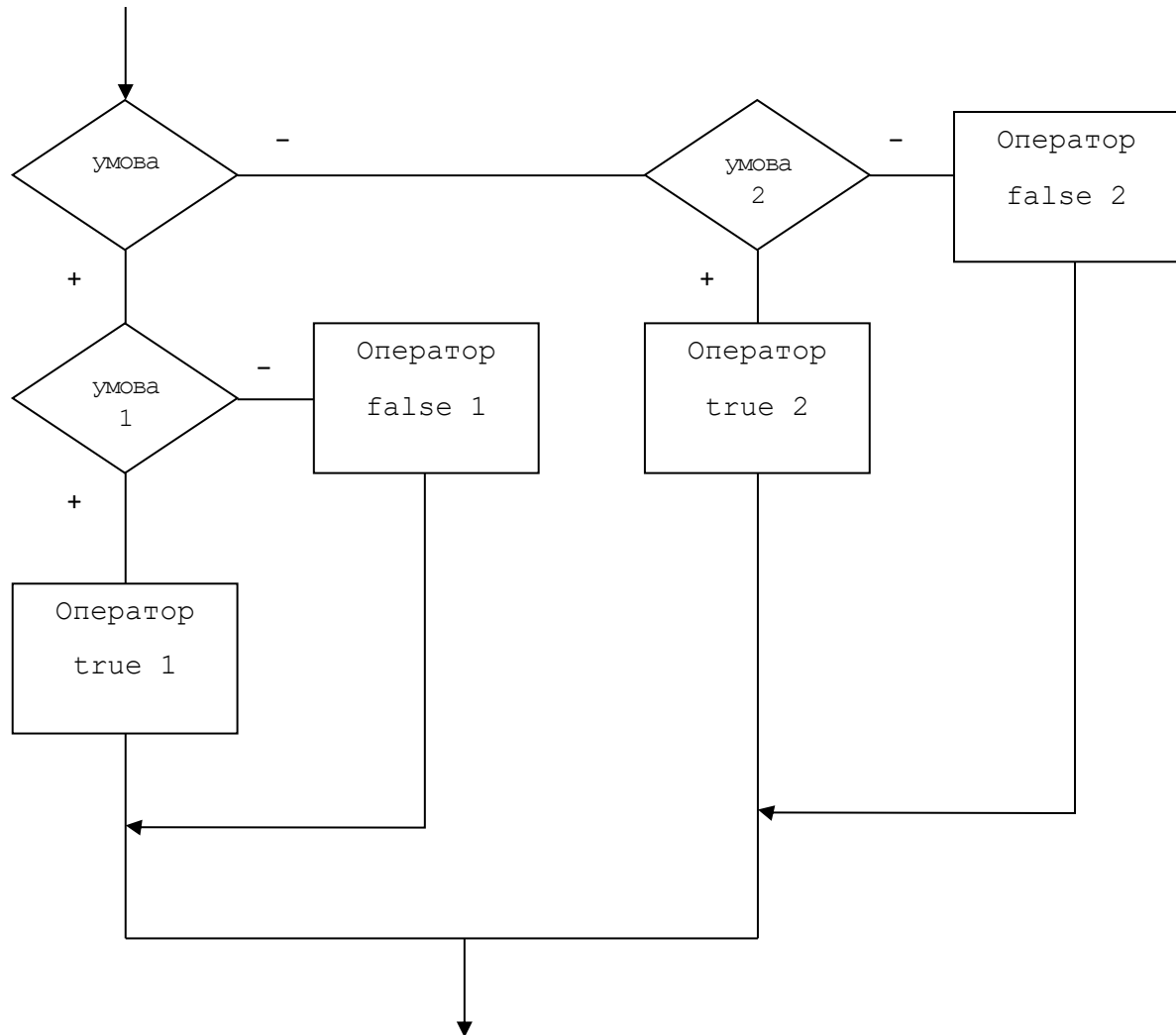
В цьому випадку, якщо вираз – умова отримує значення 0, програма переходить до виконання наступного оператора;

- за допомогою вкладених один в інший декількох операторів if – else можна реалізувати множинний вибір.
- єдиний "підводний камінь" у застосуванні вкладених умовних операторів складається у відповідності if й else.

Кожен *else* повинен відповідати найближчому *if*.

```
if (умова)
if (умова1)
оператор_true1;
else                               // ставиться к if з умовою1
оператор_false1;
else                               // ставиться к if з умовою
if (умова2)
оператор_true2;
else                               // ставиться к if з умовою2
оператор_false2;
```

На блок – схемі ця конструкція зображується наступним чином:



ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Для заданих користувачем вхідних даних скласти та налагодити програму обчислення значення функції згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- для введення значення вхідних даних використовувати функцію `scanf()`. Кожне вхідне дане вводити окремо;
- для виведення значення функції використовувати функцію `printf()`;
- враховувати область визначення функції. При введенні вхідних даних, що не входять в область визначення функції, повинно бути виведено відповідне повідомлення (наприклад: На введенных входных данных функция не определена)

– введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення, для виведення яких використовувати потік виведення cout.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

1 Постановка задачі.

2 Блок-схема алгоритму рішення задачі.

3 Текст програми мовою програмування C/C++.

4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Вхідні дані	Формули для обчислення
1	Дійсні числа d, n, g .	$f = \begin{cases} \frac{n-2}{n-d}, d > 0 \\ g + \sqrt[3]{n}, d \leq 0 \end{cases}$
2	Дійсні числа S, a, t, k .	$g = \begin{cases} \frac{a-t^3}{S}, k < \frac{S}{2} \\ t-a^4, k \geq \frac{S}{2} \end{cases}$
3	Дійсні числа s, d, k .	$f = \begin{cases} \frac{d-k}{k-d}, s = \frac{d}{k} \\ \frac{d+k}{k+d}, s \neq \frac{d}{k} \end{cases}$
4	Дійсні числа g, a, b .	$d = \begin{cases} a - \sqrt{b^3}, g > 2 \\ a + \sqrt[3]{b}, g \leq 2 \end{cases}$
5	Дійсні числа r, a, d .	$S = \begin{cases} r - \frac{d}{r}, r \neq 0 \\ r - \frac{a}{d}, r = 0 \end{cases}$

Варіант	Вхідні дані	Формули для обчислення
6	Дійсні числа s, h, a, t .	$K = \begin{cases} \frac{h-s^2}{2h}, s > 0 \\ a - \sqrt{t}, s \leq 0 \end{cases}$
7	Дійсні числа d, c, r .	$X = \begin{cases} \frac{r-2c}{d}, d > -5 \\ \frac{c-2r}{c+d}, d \leq -5 \end{cases}$
8	Дійсні числа a, s, r, x, y .	$h = \begin{cases} a + \sqrt{\frac{s}{r}}, a > 2 \\ \frac{y+x}{3a}, a \leq 2 \end{cases}$
9	Дійсні числа f, k .	$d = \begin{cases} f * \sqrt{k}, f > 0 \\ \frac{f}{\sqrt{k}}, f \leq 0 \end{cases}$
10	Дійсні числа d, b, f, k, a, y .	$p = \begin{cases} b + \frac{f}{\sqrt{k}}, d > \frac{b}{2} \\ a - \frac{3 \sin^3 y}{2+a}, d \leq \frac{b}{2} \end{cases}$
11	Дійсні числа f, g .	$d = \begin{cases} \frac{f-g^2}{f+g}, f > 0 \\ \frac{g-f^3}{g-f}, f \leq 0 \end{cases}$
12	Дійсні числа g, s, r, d, t .	$k = \begin{cases} s - 2 * \sqrt{r}, g > 5 \\ d + \frac{t}{3g}, g \leq 5 \end{cases}$
13	Дійсні числа f, g .	$N = \begin{cases} \frac{f}{g^3} - 1, f < 0 \\ \frac{g^2}{f} - 1, f \geq 0 \end{cases}$
14	Дійсні числа n, t .	$b = \begin{cases} t - 2\sqrt{n-t}, n > 2 \\ 3\sqrt{n-t}, n \leq 2 \end{cases}$
15	Дійсні числа k, f, i, g, n .	$y = \begin{cases} f * \sqrt{i}, k = 0 \\ \frac{g-n}{2-g}, k \neq 0 \end{cases}$
16	Дійсні числа p, o, f, n	$y = \begin{cases} \frac{o+p}{3o}, p \neq 2 \\ f + 3\sqrt{n}, p = 2 \end{cases}$

Варіант	Вхідні дані	Формули для обчислення
17	Дійсні числа d, s, f	$Z = \begin{cases} \frac{s}{2s-d} - 5, d > 0 \\ f - \frac{s}{3d}, d \leq 0 \end{cases}$
18	Дійсні числа s, f, k, h	$g = \begin{cases} \frac{f-k}{f+k}, s = 0 \\ f - \sqrt{2 * h}, s \neq 0 \end{cases}$
19	Дійсні числа g, s, f, n	$d = \begin{cases} s - \frac{f}{\sqrt{n}}, g > 0 \\ f - \sqrt[3]{n}, g \leq 0 \end{cases}$
20	Дійсні числа k, s, j, t	$f = \begin{cases} j + (s * \sqrt{k}), k > 0 \\ t - (2 - \sqrt{s}), k \leq 0 \end{cases}$
21	Дійсні числа g, d, f, n	$x = \begin{cases} \frac{d}{n} + \frac{n}{d}, g > 5 \\ f - \sqrt{n * 2}, g \leq 5 \end{cases}$
22	Дійсні числа g, s, f, n	$g = \begin{cases} f - \sqrt{t}, b > 0 \\ a+b + \sqrt{a-b} f, b \leq 0 \end{cases}$
23	Дійсні числа d, g, f, x	$S = \begin{cases} g + \sqrt{x}, d > 3 \\ f - \sqrt[3]{x}, d \leq 3 \end{cases}$
24	Дійсні числа f, r, n, g, v	$s = \begin{cases} r - \frac{f}{n}, f < 0 \\ g - \sqrt{v}, f \geq 0 \end{cases}$
25	Дійсні числа g, b, s, n	$f = \begin{cases} \frac{b}{s}(s-5) - 3 * \sqrt[3]{b}, g > 0 \\ s + 3 * \sqrt{n}, g \leq 0 \end{cases}$
26	Дійсні числа p, k, l	$d = \begin{cases} \frac{k-l}{p-l}, p > 0 \\ \frac{p-k}{l}, p \leq 0 \end{cases}$

КОНТРОЛЬНІ ПИТАННЯ

- 1 Наведіть повну форму коду умовного оператора if.
- 2 Виконайте загальний вигляд блок-схеми повного умовного оператора.
- 3 Наведіть скорочену форму коду умовного оператора if.

4 Виконайте загальний вигляд блок-схеми скороченого умовного оператора.

5 Що може включати "вираз-умова" в операторі if?

6 Що необхідно задіювати, якщо необхідне виконання не одного оператора, а деякого набору операторів в операторі if?

ЛІТЕРАТУРА

1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.

2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>

3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 4
з дисципліни Технології (Програмування)

Тема: Складання та налагодження розгалужених програм з використанням оператора switch

Мета: Придбати практичні навички при складанні та налагодженні програм розгалужених алгоритмів, використовуючи оператор-перемикач switch. Виробити навички організації однорівневого меню програми.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Оператор switch.

– Оператор switch - це інструмент для прийняття в програмі багатозначних рішень, що дозволяють виконувати одну з багатьох гілок програми. Альтернативні шляхи вибираються залежно від значення цілочисельного виразу – виразу в круглих дужках за ключовим словом switch. Інша частина оператора складається з гілок, що містять у фігурні дужки (відкриваюча й закриваюча фігурну дужки обов'язкові). Кожна гілка складається із ключового слова case, значення того ж типу, що й вираз після switch, двокрапки й набору з одного або більше операторів, що завершуються крапкою з комою:

```
switch (вираз)
{
    // фігурні дужки обов'язкові
    case константа_вираз1: оператори;    // перша гілка
    case константа_вираз2: оператори;    // інші гілки
    . . .
    default: оператори;                  // гілка за замовчуванням
}
```

– Вираз в операторі switch може бути тільки типу char, short, int або long (цілі типи). Типи із плаваючою крапкою (float, double або long double) не допускаються. Не можна використовувати й типи, обумовлені програмістом: структури, масиви або класи. Мітки в case повинні бути виразами-константами, тобто константами етапу компіляції, і того ж типу, що й вираз після switch.

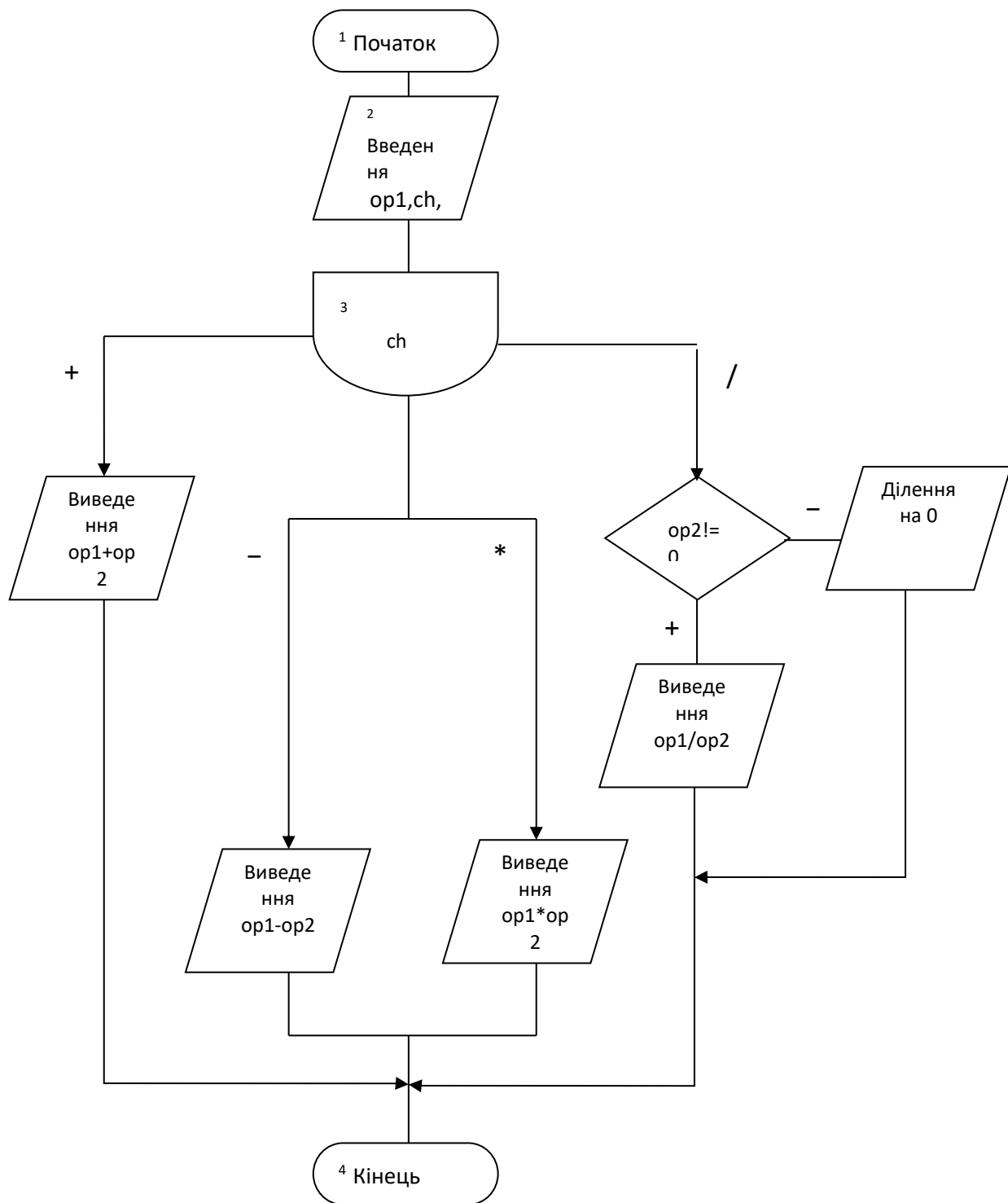
У випадку оператора `switch` дві мітки `case` не можуть бути однаковими (якщо тільки вони не ставляться до різних операторів `switch`).

– При виконанні оператора значення виразу `switch` порівнюється (зверху вниз) з літералами (мітками) `case`. Якщо значення виразу збігається з міткою, то виконання програми продовжується з операторів, що слідують за цією міткою до кінця оператора `switch`. Якщо жодна з міток не відповідає значенню виразу `switch`, то це не помилка. У такому випадку виконуються оператори, що слідують за ключовим словом `default`. Мітка `default` не обов'язкова. Звичайно вона є останньою в операторі `switch`, але може міститися й у середину оператора. Якщо вона відсутня і немає міток, що відповідають значенню виразу `switch`, то всі оператори в конструкції `switch` пропускаються й виконується наступний оператор програми. Помітимо, що оператори в `case`-гілках `switch` не потрібно брати у фігурні дужки. Вони не змінять порядок виконання - всі оператори обробляються послідовно.

– Якщо після виконання певної гілки оператора `switch` необхідно вийти з оператора використовується оператор `break`, який передає керування наступному операторові після закриваючої дужки `switch`. Для передачі керування з оператора `switch` можна використати оператори `goto` або `exit`, але в більшості випадків досить оператора `break`.

– Приклад програми – калькулятора, що виконує чотири арифметичні дії: `+`, `-`, `*` та `/` над дійсними числами. Програма пропонує користувачеві ввести значення першого операнда, знак операції та значення другого операнда.

Блок-схема алгоритму рішення задачі:



Текст програми

```
#include <iostream.h>
#include <cstdlib.h>
using namespace std;
void main(void)
{
```

```

double op1, op2; char ch;
cout << "Введите операнд, операцию и другой операнд:
cin >> op1 >>ch >>op2;
switch(ch)
{
    // обов'язкові фігурні дужки
case '+' : cout <<"Результат равен " << op1 + op2 <<endl; break;
case '*' : cout << "Результат равен " << op1 * op2 << endl; break;
case '-' : cout << "Результат равен " << op1 - op2 << endl; break;
case / :   if (op2 != 0.0) cout << "Результат равен " << op1 / op2 <<
endl;
    else cout << "Деление на ноль" << endl; break;
default: cout << "Недопустимая операция" « endl; break;
// тут break необов'язковий

}
return;
}

```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Для заданих дійсних чисел x , a , d , c розробити програму обчислення значення функції F за формулами згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- вибір користувачем однієї трьох формул, за якою проводити розрахунок, організувати за допомогою меню. Рекомендований вид екрану – меню приведено нижче. Дані, що вводить користувач, виділені напівжирним шрифтом:

Вычисление значения функции F по заданным формулам:

$$1 \ F = a * (x+7) * (x+7) - b$$

$$2 \ F = (x - c * d) / (a * x)$$

$$3 \ F = x / c$$

Выберите номер формулы – **2**

- враховувати область визначення функції. При введенні вхідних даних, що не входять в область визначення функції, повинно бути виведено відповідне повідомлення;

- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування Сі/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

1 Постановка задачі.

2 Блок-схема алгоритму рішення задачі.

3 Текст програми мовою програмування С/C++.

4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Формули обчислення	Варіант	Формули обчислення
1	$F = \begin{cases} \frac{ax^2 + b}{x - a} \\ \frac{x}{c} \end{cases}$	2	$F = \begin{cases} \frac{\frac{1}{ax} - b}{x - a} \\ \frac{10x}{c - 4} \end{cases}$
3	$F = \begin{cases} \frac{ax^2 + bx + c}{x - c} \\ a(x + c) \end{cases}$	4	$F = \begin{cases} \frac{-ax - c}{x - a} \\ \frac{bx}{c - a} \end{cases}$
5	$F = \begin{cases} \frac{a - \frac{x}{10 + b}}{x - a} \\ 3x + \frac{2}{c} \end{cases}$	6	$F = \begin{cases} \frac{ax^2 + b^2x}{x + a} \\ \frac{x}{c} \end{cases}$

Варіант	Формули обчислення	Варіант	Формули обчислення
7	$F = \begin{cases} -ax^2 - b \\ \frac{x - a}{x} \\ -\frac{x}{c} \end{cases}$	8	$F = \begin{cases} -ax^2 \\ \frac{a - x}{cx} \\ \frac{x}{c} \end{cases}$
9	$F = \begin{cases} ax^2 + b^2x \\ x - \frac{a}{x - c} \\ 1 + \frac{x}{c} \end{cases}$	10	$F = \begin{cases} ax^2 - bx + c \\ \frac{x - a}{x - c} \\ \frac{x}{c} \end{cases}$
11	$F = \begin{cases} ax^2 + \frac{b}{c} \\ \frac{x - a}{(x - c)^2} \\ \frac{x^2}{c^2} \end{cases}$	12	$F = \begin{cases} ax^2 + b^2 + c \\ \frac{x - a}{x - c} \\ \frac{x}{c} + \frac{x}{a} \end{cases}$
13	$F = \begin{cases} ax^2 + b \\ \frac{x - a}{x} \\ \frac{x}{c} \end{cases}$	14	$F = \begin{cases} -ax^3 - b \\ \frac{x - a}{x - c} \\ \frac{x}{c} + \frac{c}{x} \end{cases}$
15	$F = \begin{cases} -ax^2 + b \\ \frac{x}{x - c} + 5.5 \\ \frac{x}{-c} \end{cases}$	16	$F = \begin{cases} a(x + c)^2 - b \\ \frac{x - a}{-c} \\ a + \frac{x}{c} \end{cases}$
17	$F = \begin{cases} ax^2 - cx + b \\ \frac{x - a}{x - c} \\ -\frac{x}{a - c} \end{cases}$	18	$F = \begin{cases} ax^3 + bx^2 \\ \frac{x - a}{x - c} \\ \frac{x + 5}{c(x - 10)} \end{cases}$

Варіант	Формули обчислення	Варіант	Формули обчислення
19	$F = \begin{cases} \frac{a(x+7)^2 - b}{x - cd} \\ \frac{ax}{x} \\ \frac{x}{c} \end{cases}$	20	$F = \begin{cases} -\frac{2x - c}{cx - a} \\ \frac{x - a}{x - c} \\ -\frac{x}{c} + \frac{-c}{2x} \end{cases}$
21	$F = \begin{cases} \frac{-ax^2 - b}{x - a} \\ \frac{x}{x} \\ \frac{-x}{c} \end{cases}$	22	$F = \begin{cases} -ax^2 \\ \frac{a - x}{cx} \\ \frac{x}{c} \end{cases}$
23	$F = \begin{cases} \frac{ax^2 + b^2x}{x - \frac{a}{x - c}} \\ 1 + \frac{x}{c} \end{cases}$	24	$F = \begin{cases} \frac{ax^2 - bx + c}{x - a} \\ \frac{x - a}{x - c} \\ \frac{x}{c} \end{cases}$
25	$F = \begin{cases} \frac{ax^2 + \frac{b}{c}}{x - a} \\ \frac{(x - c)^2}{x^2} \\ \frac{x^2}{c^2} \end{cases}$	26	$F = \begin{cases} \frac{ax^2 + b^2 + c}{x - a} \\ \frac{x - a}{x - c} \\ \frac{x}{c} + \frac{x}{a} \end{cases}$
27	$F = \begin{cases} \frac{ax^2 + b}{x - a} \\ \frac{x}{x} \\ \frac{x}{c} \end{cases}$	28	$F = \begin{cases} -ax^3 - b \\ \frac{x - a}{x - c} \\ \frac{x}{c} + \frac{c}{x} \end{cases}$

КОНТРОЛЬНІ ПИТАННЯ

- 1 Опишіть призначення оператора switch.
- 2 Опишіть загальну форму запису оператора switch.
- 3 Опишіть правила виконання оператора switch.

- 4 Якого типу може бути константний вираз для порівняння з заданими варіантами у операторі switch?
- 5 Для чого призначене ключове слово case в операторі switch?
- 6 Для чого в операторі switch використовується оператор break?
- 7 Чи буде порушена логіка роботи програми в разі відсутності оператора break в гілках оператора switch?
- 8 Опишіть призначення гілки default в операторі switch.
- 9 З яких символів можуть складатися імена міток?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 5
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм з використанням ітераційних циклів

Мета: Придбати практичні навички при складанні та налагодженні циклічних програм з використанням операторів циклів з передумовою `while` та постумовою `do - while`.

Час 2 години

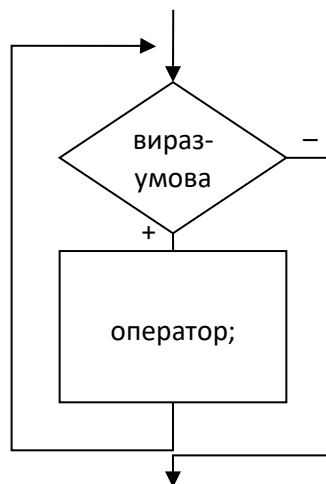
ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 `while` – цикл з передумовою

– Цикл `while` має наступну форму запису:

```
while (вираз)           // умова виходу з циклу
оператор;               // тіло циклу
```

В блок – схемах оператор циклу з передумовою зображується наступним чином:



– Дана керуюча структура повторює тіло циклу доти, поки його вираз – умова виходу з циклу приймає значення, відмінне від 0 (або значення `true`, якщо вираз – умова є логічним). Як тільки вираз – умова стане рівним 0 (або `false`), тіло циклу пропускається, оператор `while` припиняє свою роботу і починає виконуватись оператор, наступний за `while`.

– Якщо тіло циклу складається з декількох операторів, необхідно використовувати обмежники блоку (фігурні дужки):

```
while (вираз)
{
Оператор_1;
.
.
Оператор_n;
}
```

– Для завершення циклу `while` у тілі циклу обов'язково повинні бути інструкції, виконання яких впливає на умову завершення циклу;

– Цикл `while` - це цикл із передумовою, а тому можлива ситуація, при якій інструкції тіла циклу жодного разу не будуть виконані, наприклад:

```
k=5;
n=10;
while (k<n)
{
printf("k=%d n=%d\n", до, n);
k+=2;
k=k+2;
n++;
}
```

2 do-while – цикл з постумовою

– Цикл `do-while` дуже нагадує цикл `while`. Часто вони взаємозамінні. Основна різниця в тім, що в циклі `do-while` умова виходу з циклу перевіряється наприкінці циклу, після кожної ітерації, а в циклі `while` вона перевіряється на початку циклу, перед ітерацією.

– Як і цикл `while`, цикл `do-while` управляє повторюваним виконанням тіла циклу, що складає з одного оператора (або блоку операторів у фігурних дужках).

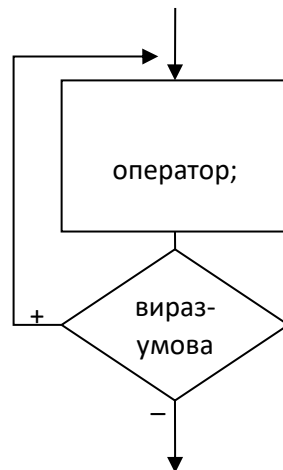
– Цикл `do-while` має загальну структуру:

```
do
```

```
оператор;                // або { оператори }  
while (вираз);
```

Така структура забезпечує як мінімум однократне виконання тіла циклу. Щоб завершити цикл і запобігти нескінченним ітераціям, у тілі циклу повинен змінюватися вираз циклу.

В блок – схемах оператор циклу з постумовою зображується наступним чином:



- Цикл `do - while`, як правило, використовується для організації наближених обчислень, у завданнях пошуку й обробки даних, що вводять із клавіатури або з файлу.

- Щоб уникнути плутанини програмісти часто містять тіло циклу у фігурні дужки, навіть якщо воно містить лише один оператор:

```
do  
{ оператор; }  
while (вираз);
```

- Щоб уникнути синтаксичної помилки, у циклі `do-while` після виразу – умови потрібно поставити крапку з коми (на відміну від циклу `while`, де вона саме не потрібна). Варто розуміти, що розміщення крапки з комою після виразу – умови в циклі `while` не бентежить компілятор і не приводить до синтаксичної помилки, але код стає некоректним (семантична помилка). Щоб указати на особливий зміст ключового слова `while` і на необхідність використання крапки з комою наприкінці рядка, деякі програмісти розміщують закриваючу фігурну дужку на той же рядок, де перебуває ключове слово `while`.

```
do
{ оператор;
} while (вираз); //фігурна дужка попереджає про присутність
// крапки з комою
```

– Цикли `while` або `do - while` доцільно використовувати для забезпечення коректного введення користувачем вхідних даних. Якщо вводяться дійсні або цілі числа, використовуються також стандартні функції `atof` та `atoi`, визначені в заголовному файлі `<cstdlib.h>`

1. Функція **atof**

```
#include <cstdlib.h>
double atof(const char *str);
```

Функція `atof()` перетворює рядок символів `str` в значення типу `double`. Рядок повинен містити коректне представлення дійсного числа з плаваючою крапкою. Інакше значення, що повертається, не визначено (тобто дорівнює `NULL`)

Число може завершуватись будь-яким символом, недопустимим для представлення дійсних чисел: роздільником, знаком пунктуації (але не крапкою), символом, відмінним від букв `E` та `e`.

2. Функція **atoi**

```
#include <cstdlib.h>
int atoi(const char *str);
```

Функція `atoi()` перетворює рядок символів `str` в значення типу `int`. Рядок повинен містити коректне представлення цілого числа. Інакше значення, що повертається, не визначено (тобто дорівнює `NULL`)

Число може завершуватись будь-яким символом, недопустимим для представлення цілих чисел: роздільником, знаком пунктуації, символом.

Приклад фрагменту програми забезпечення коректного введення цілого числа.

```
#include <iostream.h>
#include <stdio.h>
#include <cstdlib.h>
int x;
char p[12];
. . .
cout<<"Введіть ціле число"<<endl;
gets(p);
while(atoi(p)==NULL)
```

```

{
    cout<<"Некорректный ввод. Вводите только целое число"<<endl;
    gets (p) ;
}
p = atoi (p) ;

```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Використовуючи оператор циклу з передумовою while або оператор циклу з постумовою do...while скласти програму рішення задачі згідно Вашого варіанта. Програма повинна задовольняти наступним вимогам:

- для зберігання введеного числа використовувати просту змінну відповідного типу;
- враховувати, що користувач може не ввести жодного шуканого числа. В цьому випадку необхідно вивести відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

- 1 Підраховувати суму додатних чисел серед тих дійсних чисел, що вводяться користувачем, поки користувач не введе 0.

- 2 Підраховувати добуток від'ємних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено непарне число.
- 3 Підраховувати суму додатних чисел серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число 100.
- 4 Підраховувати кількість нулів серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число 50.
- 5 Підраховувати добуток від'ємних чисел серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число 0.
- 6 Підраховувати добуток додатних чисел серед тих дійсних чисел, що вводяться користувачем, поки добуток не перевищить 260.
- 7 Підраховувати суму парних чисел, серед тих цілих чисел, що вводяться користувачем, поки не буде введено число 0.
- 8 Підраховувати середнє арифметичне чисел серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число -1.
- 9 Підраховувати середнє арифметичне парних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено число 51.
- 10 Підраховувати середнє арифметичне додатних чисел серед тих дійсних чисел, що вводяться користувачем, поки не буде введено непарне число.
- 11 Підраховувати кількість та суму ненульових чисел серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число 15.
- 12 Підраховувати суму тих чисел, що належать діапазону $[0;5]$, серед дійсних чисел, що вводяться користувачем, поки сума не перевищить 15.
- 13 Підраховувати середнє арифметичне від'ємних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено парне число.
- 14 Підраховувати кількість тих чисел, що належать діапазону $[-10;+10]$ серед дійсних чисел, що вводяться користувачем, поки не буде введено число 5.
- 15 Підраховувати суму невід'ємних чисел з діапазону $[-11;+15]$ серед тих цілих чисел, що вводяться користувачем, поки не буде введено непарне число.
- 16 Підраховувати середнє арифметичне від'ємних чисел з діапазону $[9;41]$ серед тих дійсних чисел, що вводяться користувачем, поки середнє арифметичне не перевищить 15.
- 17 Підраховувати кількість парних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено число більше 20.
- 18 Підраховувати добуток від'ємних чисел з діапазону $[0;15]$ серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число менше -10.
- 19 Підраховувати кількість нулів серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число з діапазону $[7;11]$.
- 20 Підраховувати середнє арифметичне непарних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено число 100.

- 21 Підраховувати добуток чисел з діапазону $[-2;4]$ серед тих дійсних чисел, що вводяться користувачем, поки не буде введено число 0.
- 22 Підраховувати середнє арифметичне непарних чисел і серед тих цілих чисел, що вводяться користувачем, поки не буде введено число з діапазону $[-1;3]$.
- 23 Підраховувати кількість невід'ємних цілих чисел з діапазону $[-10;+11]$ серед тих цілих чисел, що вводяться користувачем, поки не буде введено парне число.
- 24 Підраховувати суму непарних чисел серед тих цілих чисел, що вводяться користувачем, поки сума не перевищить 14.
- 25 Підраховувати суму та добуток непарних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено число 0.
- 26 Підраховувати середнє арифметичне парних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено число -17.
- 27 Підраховувати середнє арифметичне непарних чисел серед тих цілих чисел, що вводяться користувачем, поки не буде введено число з діапазону $[-4;0]$.
- 28 Підраховувати добуток парних чисел з діапазону $[-3;16]$ серед тих цілих чисел, що вводяться користувачем, поки не буде введено число 90.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Назвіть типи циклів.
- 2 Що називається тілом циклу?
- 3 Що називається кроком циклу?
- 4 Наведіть загальний вигляд оператора циклу з передумовою.
- 5 Який оператор відповідає циклу з передумовою?
- 6 Наведіть загальний вигляд оператора циклу з постумовою.
- 7 Який оператор відповідає циклу з постумовою?
- 8 Які мінімальну кількість разів виконується цикл while?
- 9 Які мінімальну кількість разів виконується цикл do-while?
- 10 Які необхідні умови правильної організації циклів while та do-while?
- 11 Для чого використовуються та як працюють стандартні функції atof та atoi?

Методичні рекомендації
до виконання лабораторної роботи № 6
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм з використанням регулярних циклів.

Мета: Придбати практичні навички при складанні та налагодженні циклічних програм з використанням оператора циклу `for`.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Регулярний цикл `for`

– Цикл `for` підходить у тих випадках, коли число ітерацій відомо заздалегідь, до початку циклу. Цикл `for` має наступну стандартну форму, де в дужках комбінуються три основні вирази, що управляють виконанням тіла циклу (ініціалізація поточних даних, їхнє обчислення й модифікація). *Ці вирази розділяються двокрапками:*

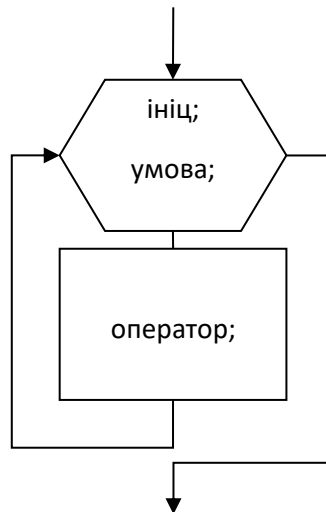
```
for (ініціалізація; вираз – умова; корекція) // заголовок циклу  
оператор; \\ тіло циклу
```

Ініціалізація (або початковий вираз) обчислюється тільки один раз, перед першою ітерацією. Тут зручно ініціалізувати значення для роботи циклу: індекс, лічильник елементів, суму й ін.

Вираз – умова обчислюється перед першою ітерацією й перед кожною наступною ітерацією. Даний вираз оцінити необхідність наступної ітерації циклу. Якщо при його обчисленні виходить значення, відмінне від 0 (або `true`), то виконується оператор циклу. Якщо ж у результаті обчислення виразу – умови буде отримане значення 0 (або `false`), то виконання оператора циклу буде завершено.

Корекція обчислюється наприкінці наступної ітерації, безпосередньо після виконання тіла циклу. Це загальноприйняте місце зміни поточних даних, збільшення індексів, лічильників, міток і т.д.

На блок – схемі регулярний цикл `for` зображується наступним чином:



Потрібно розуміти, що цикл `for` еквівалентний наступному циклу `while`:

```
ініціалізація;  
while (вираз - умова)  
{  
    оператор;  
    корекція;  
}
```

- Кожна із трьох складових у заголовку циклу `for` являє собою вираз.

Це означає, що в кожному такому виразі може використатися послідовність розділених комами виразів. Не забувайте, що кома - повноправна операція C++. Її операнди (вирази) обчислюються зліва направо і повертається саме праве значення. Отже, в виразі ініціалізації можна задавати початкові значення декільком змінним, описувати ці змінні, якщо вони відносяться до одного типу. Вираз – корекція також може складатись із декількох виразів (наприклад, виразів присвоювання). Увага! *Вираз – умова може складатись тільки з одного виразу.*

Приклад фрагменту програми обчислення суми квадратів перших `n` натуральних чисел:

```
int main ()  
{  
    int num;  
    cout << "\n Введіть, скільки нужно сложить квадратов: ";  
    cin >> num;  
    for (int sum = 0, n = 1; n <= num; sum+=n*n, n++);  
    cout << "Сумма квадратов равна " << sum << endl;  
    return 0;  
}
```

Зверніть увагу! В цьому прикладі тіло циклу `for` – пустий оператор.

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Використовуючи оператор циклу for скласти програму рішення задачі згідно Вашого варіанта. Програма повинна задовольняти наступним вимогам:

- для зберігання введеного числа використовувати просту змінну відповідного типу;
- враховувати, що користувач може не ввести жодного шуканого числа. В цьому випадку необхідно вивести відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

- 1 Порахувати суму додатних чисел із 10-ти дійсних чисел, що вводяться користувачем.
- 2 Порахувати добуток парних чисел із 15-ти цілих чисел, що вводяться користувачем.
- 3 Порахувати суму від'ємних чисел з 10-ти дійсних чисел, що вводяться користувачем.
- 4 Порахувати кількість нулів серед 13-ти дійсних чисел, що вводяться користувачем.
- 5 Порахувати добуток від'ємних чисел із 7-ми дійсних чисел, що вводяться користувачем.

- 6 Порахувати добуток додатних чисел із 8-ми дійсних чисел, що вводяться користувачем.
- 7 Порахувати суму парних чисел із 10-ти цілих чисел, що вводяться користувачем.
- 8 Порахувати середнє арифметичне 10-ти чисел дійсних чисел, що вводяться користувачем.
- 9 Порахувати середнє арифметичне парних чисел із 12-ти цілих чисел, що вводяться користувачем.
- 10 Порахувати середнє арифметичне додатних чисел серед 8-ми дійсних чисел, що вводяться користувачем.
- 11 Порахувати кількість ненульових чисел із 15-ти дійсних чисел, що вводяться користувачем.
- 12 Серед 9-ти дійсних чисел, що вводяться користувачем порахувати суму чисел з діапазону $[0;5]$
- 13 Порахувати середнє арифметичне від'ємних чисел із 7-ми дійсних чисел, що вводяться користувачем.
- 14 Порахувати кількість чисел з діапазону $[-10;+10]$ серед 9-ти дійсних чисел, що вводяться користувачем.
- 15 Порахувати суму невід'ємних парних чисел з діапазону $[5;+15]$ серед 9-ти цілих чисел, що вводяться користувачем.
- 16 Порахувати середнє арифметичне від'ємних непарних чисел з діапазону $[-30;-10]$ серед 9-ти цілих чисел, що вводяться користувачем.
- 17 Порахувати кількість парних чисел серед 11-ти цілих чисел, що вводяться користувачем.
- 18 Порахувати добуток від'ємних чисел з діапазону $[-15, -5]$ серед 9-ти дійсних чисел, що вводяться користувачем.
- 19 Порахувати кількість нулів серед 11-ти чисел дійсних чисел, що вводяться користувачем.
- 20 Обчислити середнє арифметичне непарних чисел із 9-ти цілих чисел, що вводяться користувачем.
- 21 Порахувати добуток чисел з діапазону $[-2;4]$ серед 10-ти дійсних чисел, що вводяться користувачем.
- 22 Обчислити середнє арифметичне непарних чисел із 12-ти цілих чисел, що вводяться користувачем.
- 23 Порахувати кількість невід'ємних чисел з діапазону $[3; 11]$ серед 9-ти дійсних чисел, що вводяться користувачем.
- 24 Порахувати суму непарних чисел із 11-ти цілих чисел, що вводяться користувачем.

- 25 Порахувати суму та добуток непарних чисел із 12-ти цілих чисел, що вводяться користувачем.
- 26 Обчислити середнє арифметичне парних чисел із 18-ти цілих чисел, що вводяться користувачем.
- 27 Обчислити середнє арифметичне непарних чисел із 13-ти цілих чисел, що вводяться користувачем.
- 28 Порахувати добуток парних чисел з діапазону $[-3;16]$ серед 9-ти цілих чисел, що вводяться користувачем.
- 29 Із 10-ти дійсних чисел, що вводяться користувачем порахувати добуток тих чисел, що належать інтервалу $[-20, 10]$.
- 30 Обчислити середнє арифметичне непарних чисел та середнє арифметичне додатних із 12-ти цілих чисел, що вводяться користувачем.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Які цикли називаються регулярними?
- 2 Яка загальна форма запису (синтаксис) циклу for?
- 3 З яких елементів складається заголовок циклу for?
- 4 Яке призначення виразу – ініціалізації?
- 5 Що можна використовувати в виразі – ініціалізації?
- 6 Яке призначення виразу – умови?
- 7 Яке призначення виразу – корекції?
- 8 Що можна використовувати в виразі – корекції?
- 9 Описати правила роботи (семантику) оператора циклу for.
- 10 Чи може в операторі циклу for бути відсутній будь-який з трьох виразів в заголовку?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 7
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм обробки одномірних масивів

Мета: Придбати практичні навички опису, введення – виведення та обробки лінійних (одномірних) масивів.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Поняття масиву

– Масив - це набір елементів даних одного типу. Їхній стан характеризується набором значень, а не одним значенням. До кожного значення - компоненту масиву можна звертатися за допомогою спеціального запису C++ (операції індексу);

– Масиви - упорядковані сукупності даних. Це означає, що кожен елемент масиву має попередній і наступний елемент, хоча є два очевидних виключення: перший елемент масиву не має попереднього, а останній - наступного. Масив має ім'я, але окремі елементи масиву імен не мають. Програма звертається до них по імені з додаванням індексу, що визначає позицію елемента в упорядкованому наборі;

– Масиви кінцеві. Число елементів масиву повинне бути відомо під час компіляції й не може змінюватися при виконанні програми.

–

2 Визначення масивів у C++

– Як і будь-якого змінна C++ , змінна типу масиву повинна попередньо визначатися й лише потім використатися. Визначення масиву зв'язує ім'я масиву з елементами масиву й числом елементів. Як і будь-яке визначення, визначення масиву приводить до виділення пам'яті під час виконання, як й інші визначення, воно закінчується крапкою з комою. Масиви можна визначати по одному на кожному рядку або комбінувати на одному рядку кілька визначень, наприклад:

```
int hours[7]; char grade[35]; double amount[20];
```


На цьому рядку визначаються три масиви: масив `hours[]` з 7 цілочисленних компонентів, масив `gracle[]` з 35 символьних компонентів і масив `amount[]` з 20 компонентів із плаваючою крапкою подвійної точності.

– У випадку масивів різних типів (як у попередньому прикладі) кожен масив потрібно визначати окремо, завершуючи визначення крапкою з комою. Для масивів одного типу можна визначити кілька масивів через кому, завершивши останній крапкою з коми. Допускається комбінування визначень масивів і скалярних типів, якщо їхні типи збігаються, наприклад:

```
int category[7], i, num, scores[35], n;
```

– Розмір масиву повинен бути відомий під час компіляції, але він не обов'язково задається літеральним значенням. Можна використати визначений в `#define` символьний літерал, цілу константу або цілий вираз будь-якої складності. Єдина вимога полягає в тому, щоб цей вираз обчислювався на етапі компіляції, а не на етапі виконання.

Наприклад:

```
#define MAX_RATES 35
// розмір масиву визначений як значення #define
int const NUM_ITEMS = 10;
// розмір масиву - константа
int rates [MAX_RATES]; double amount[2*NUM_ITEMS];
```

– Масив може ініціалізуватись (отримувати початкові значення) у визначенні аналогічно будь-якій іншій змінній C++. Програміст підставляє початкові значення - як при ініціалізації скалярних змінних. Ці початкові значення задаються в списку значень у фігурних дужках. Оскільки коми є тут роздільниками, а не завершують список, останнє значення коми не має.

```
int hours[7]={ 8, 8, 12, 8. 4, 0, 0 }; // 7 значень
int side[5]=(40,35,41 ); // 4-й та 5-й елементи масиву отримують
значення 0
char option[2]={ *Y', 'N*', 'y',    }; // синтаксична помилка
int week[52] = { , , 40, 48 }; // синтаксична помилка
```

2 Операції з масивами. Приклад програми обробки лінійного масиву

– Всі операції можна виконувати тільки з окремими елементами масиву. У цих операціях на окремі елементи масиву посилаються за допомогою операції індексу. Наприклад, `sicle[2]` позначає елемент масиву `side` з індексом 2. У кожному разі `sicle[2]` - звичайна скалярна ціла змінна. Оскільки `sicle[]` - це

масив цілих значень, з `side[2]` можна робити все те, що дозволяється робити із цілою змінною. Розходження тільки в імені - замість ідентифікатора змінної використається ім'я масиву, плюс індекс й операція індексу:

```
side[2] = 40;  
num = side[2] * 2;
```

На першому рядку `side[2]` одержує значення 40, що зберігається по відповідній адресі. На другий - значення за адресою `side[2]` множиться на 2 і результат зберігається в змінної `num` (вона повинна бути числовий). Як видно, окремі елементи масиву не мають своїх імен. Їх імена складаються з імені масиву й значення індексу.

– ***Приклад програми обробки лінійного масиву даних:***

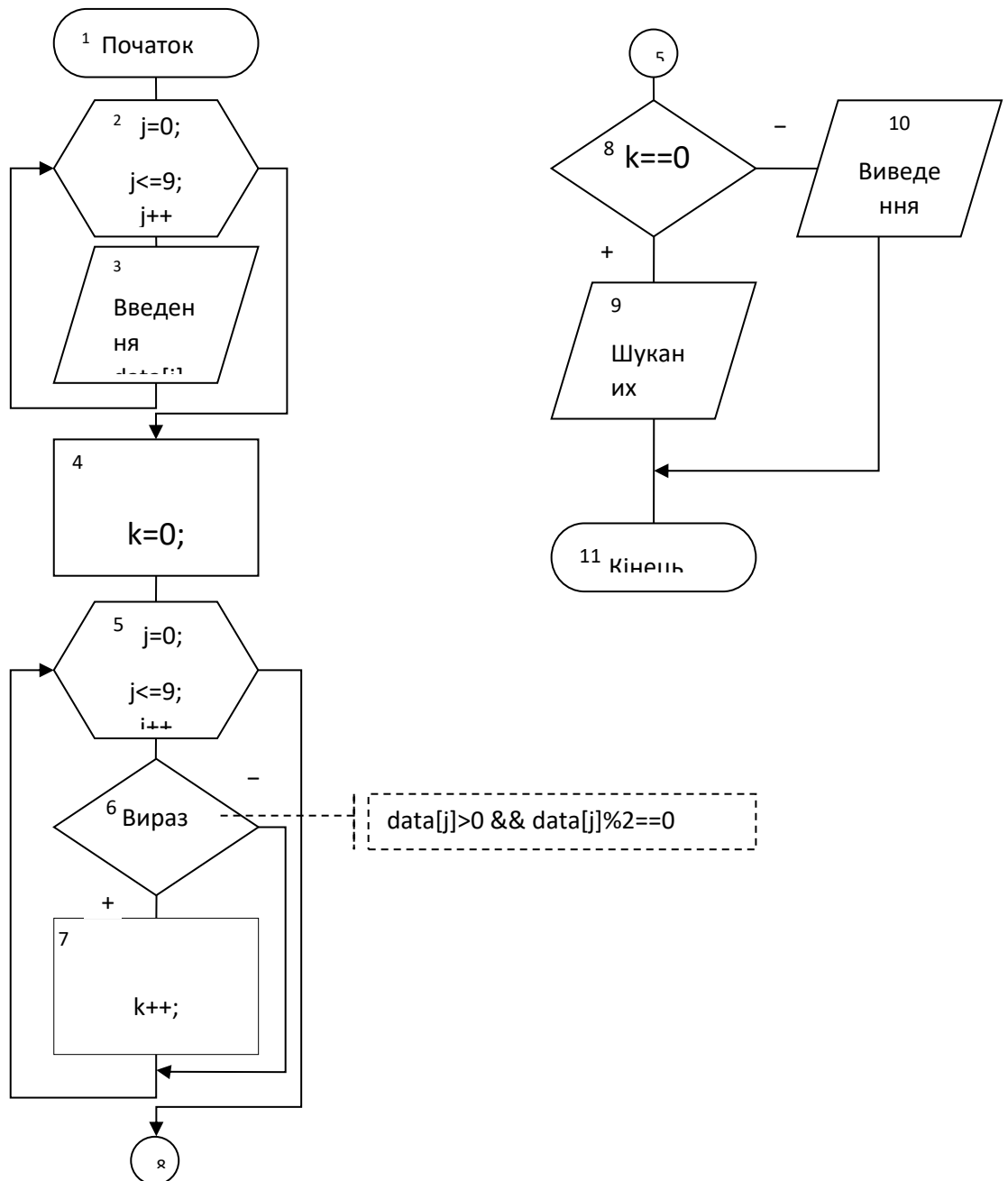
1. Постановка задачі.

В заданому масиві `data[]` із десяти цілих чисел підрахувати кількість парних додатних елементів.

Програма буде складатись з трьох блоків:

- введення значень елементів масиву;
- підрахування кількості парних додатних елементів;
- виведення результату.

2 Блок – схема алгоритму рішення задачі:



2 Текст програми

```

#include <stdio.h>
int main()
{
    int data[10];
    int j,k=0;
    for (j=0; j<=9; j++)
    {
        printf("\n Введіть елемент масива data[%i]",j);

```

```

        scanf("%i", &data[j]);
    }

    for (j=0; j<=9; j++)
    {
        if (data[j]>0 && data[j]%2==0) k++;
    }
    if (k==0)
        printf("\n Чётных положительных элементов в массиве нет");
    else
        printf("\n Количество чётных положительных в массиве data[] равно
%i", k);
    return 0;
}

```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Скласти та налагодити програму введення одновимірною (лінійного) масиву та обробки його за алгоритмом згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- введення значень елементів масиву проводити в окремому циклі;
- вивести в рядок елементи введеного масиву;
- при обробці масиву враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють всі можливі варіанти роботи програми та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Вхідні дані	Завдання на обробку
1	A(6) – масив дійсних чисел	Знайти добуток тих елементів масиву, значення яких належать діапазону [0;5].
2	B(7) – масив дійсних чисел	Знайти суму тих елементів масиву, значення яких належать діапазону [-2;+2].
3	V(8) – масив дійсних чисел	Знайти середнє арифметичне тих елементів масиву, значення яких належать діапазону [-1;3] .
4	M(9) – масив дійсних чисел	Порахувати кількість тих елементів масиву, значення яких належать діапазону [2;+5].
5	A(7) – масив цілих чисел	Знайти добуток парних елементів масиву.
6	H(6) – масив цілих чисел	Знайти суму парних елементів масиву.
7	P(8) – масив цілих чисел	Знайти середнє арифметичне парних елементів масиву.
8	L(9) – масив цілих чисел	Порахувати кількість парних елементів масиву.
9	T(6) – масив дійсних чисел	Порахувати суму додатних елементів масиву.
10	N(5) – масив дійсних чисел	Порахувати добуток додатних елементів масиву.
11	A(9) – масив дійсних чисел	Порахувати середнє арифметичне додатних елементів масиву.
12	X(6) – масив дійсних чисел	Порахувати кількість додатних елементів масиву.
13	K(6) – масив дійсних чисел	Порахувати середнє арифметичне від’ємних елементів масиву.
14	D(6) – масив дійсних чисел	Знайти суму від’ємних елементів масиву.
15	R(7) – масив дійсних чисел	Знайти добуток від’ємних елементів масиву.
16	S(5) – масив дійсних чисел	Порахувати кількість від’ємних елементів масиву.

Варіант	Вхідні дані	Завдання на обробку
17	H(9) – масив дійсних чисел	Знайти суму ненульових елементів масиву.
18	M(8) – масив дійсних чисел	Знайти середнє арифметичне ненульових елементів масиву.
19	R(9) – масив дійсних чисел	Порахувати кількість ненульових елементів масиву.
20	F(6) – масив дійсних чисел	Знайти добуток ненульових елементів масиву.
21	Z(8) – масив цілих чисел	Знайти добуток непарних елементів масиву.
22	C(7) – масив цілих чисел	Знайти суму непарних елементів масиву.
23	V(9) – масив цілих чисел	Знайти середнє арифметичне непарних елементів масиву.
24	M(8) – масив цілих чисел	Порахувати кількість непарних елементів масиву.
25	W(7) – масив цілих чисел	Знайти добуток парних додатних елементів масиву.
26	Y(6) – масив цілих чисел	Знайти суму парних додатних елементів масиву.
27	P(9) – масив цілих чисел	Знайти середнє арифметичне парних додатних елементів масиву.
28	A(6) – масив цілих чисел	Порахувати кількість парних додатних елементів масиву.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Дайте визначення поняття "масив".
- 2 Що називається розміром масиву?
- 3 Як розташовані у пам'яті елементи масиву?
- 4 Що спільного мають всі елементи масиву?
- 5 Чим відрізняються всі елементи масиву?
- 6 Наведіть загальну форму оголошення масиву.
- 7 Що називається індексом елемента масиву?
- 8 З якого номера починається нумерація елементів в масиві?

- 9 Назвіть способи ініціалізації масиву.
- 10 Чи можна змінювати розмір масиву у пам'яті під час роботи програми?
- 11 Що необхідно вказати, щоб звернутися до елементу масиву?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 8
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм пошуку екстремумів лінійних масивів даних.

Мета: Придбати практичні навички в:

- організації псевдодинамічних масивів;
- реалізації алгоритмів пошуку екстремумів лінійних масивів даних.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Формування псевдодинамічних масивів.

При описі масиву в програмі треба обов'язково вказувати кількість елементів масиву для того, щоб компілятор виділив під цей масив потрібну кількість пам'яті. Це не завжди буває зручно, оскільки число елементів в масиві може змінюватися залежно від вирішуваного завдання. В цьому випадку в програмі формують псевдодинамічні масиви. Псевдодинамічні масиви реалізуються таким чином:

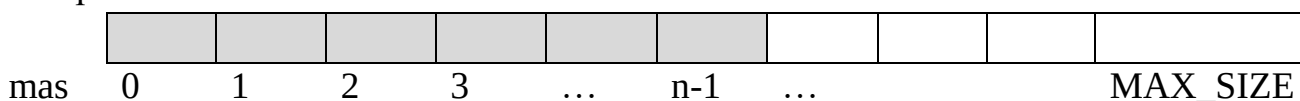
1) при визначенні масиву виділяється достатньо велика кількість пам'яті:

```
const int MAX_SIZE=100; //іменована константа
int mas[MAX_SIZE];
```

2) користувач вводить реальну кількість елементів масиву, але меншу за MAX_SIZE.

```
int n;
cout<<"\n Введіть розмір масиву, меншу за "<<MAX_SIZE<<": ";
cin>>n;
```

3) подальша робота з масивом обмежується заданою користувачем розмірністю n.



2 Алгоритм пошуку екстремумів

Знаходження екстремумів полягає у знаходженні максимального та/або мінімального елемента в масиві.

Одним з класичних алгоритмів знаходження екстремумів є наступний. Спочатку за максимальний (мінімальний) елемент береться будь-який (наприклад, нульовий) елемент масиву та заноситься в змінну «поточного» максимуму (мінімуму). Потім, за допомогою циклу від першого до останнього елементу масиву необхідно порівнювати змінну «поточного» максимуму (мінімуму) з елементами масиву і, якщо черговий елемент масиву більше (менше) «поточного» максимуму (мінімуму), необхідно зберігати цей елемент в відповідній змінній. Після проходження всього масиву в змінній «поточного» максимуму (мінімуму) буде міститись шуканий елемент.

Як бачимо, під масив `mas[]` виділяється блок пам'яті фіксованого розміру, що вміщує `MAX_SIZE` цілих чисел. А кожного разу масив обробляється до елемента з індексом `n-1`. Тому цей масив і називається псевдодинамічним.

В якості прикладу розглянемо програму пошуку останніх мінімального та максимального елементів лінійного масиву.

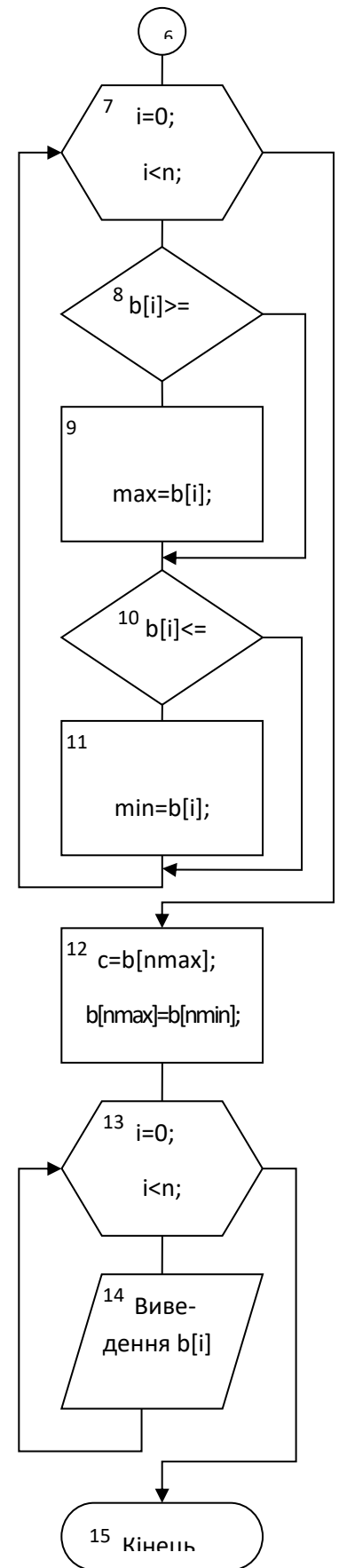
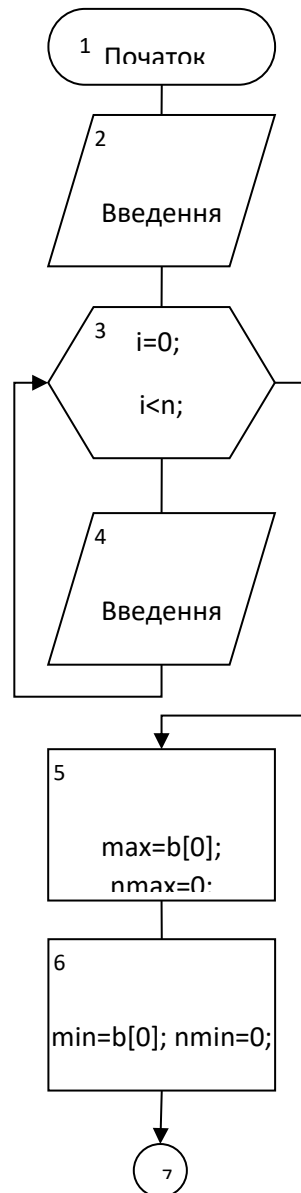
1. Постановка задачі

В заданому масиві дійсних чисел розміру `n` (`n` вводить користувач) знайти останні за розташуванням максимальний та мінімальний елементи та поміняти їх місцями.

Програма рішення задачі буде вміщати п'ять етапів:

- введення елементів масиву;
- пошук останніх за розташуванням екстремумів;
- перестановка місцями знайдених екстремумів;
- виведення оновленого масиву.

2. Блок – схема алгоритму рішення задачі



3. Текст програми.

```
#include <stdio.h>
#include <conio.h>
#include <iostream.h>
int main()
{
float b[100];
int n;
printf("\n Введіть кількість елементів в масиві");
scanf("%i", &n);
float max,min; //«Поточні» максимум та мінімум
int nmax,nmin; //Індекси максимального та мінімального елементів
int i;
//Цикл введення елементів масиву
for (i=0; i<n; i++)
{
printf("\n Введіть елемент масива b[%i] ",i+1);
cin>>b[i];
}
//Стартові значення максимуму та мінімуму та їх індексів
max=b[0]; nmax=0;
min=b[0]; nmin=0;
//Цикл пошуку екстремумів та їх індексів
for (i=1; i<n; i++)
{
if (b[i]>=max) {max=b[i]; nmax=i;}//знаки >= та <= забезпечать
пошук
if (b[i]<=min) {min=b[i]; nmin=i;}//останніх за розташуванням
екстремумів
}
//Обмін місцями максимуму та мінімуму
float c=b[nmax];//c виконує роль «буферної» змінної
b[nmax]=b[nmin];
b[nmin]=c;
//Цикл виведення оновленого масиву
printf("\n Обновленный массив");
for (i=0; i<n; i++)
cout<<endl<<b[i];
printf("\n ив");
getch();
return 0;
}
```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Дано лінійний масив з n елементів (n задає користувач). Обробити масив за алгоритмом згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- сформувати псевдодинамічний масив;
- введення значень елементів масиву проводити в окремому циклі;
- при обробці масиву враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

1 Постановка задачі.

2 Блок-схема алгоритму рішення задачі.

3 Текст програми мовою програмування C/C++.

4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють всі можливі варіанти роботи програми та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

1 В веденому масиві цілих чисел знайти та вивести мінімум парних елементів масиву. Нульові елементи масиву замінити на знайдений мінімум або на введене користувачем число в разі відсутності в масиві парних елементів. Вивести оновлений масив.

2 В веденому масиві дійсних чисел знайти та вивести мінімум додатних елементів масиву. Збільшити всі елементи масиву на знайдений мінімум або на введене користувачем число в разі відсутності додатних елементів. Вивести оновлений масив.

3 В веденому масиві дійсних чисел знайти різницю між мінімумом від'ємних елементів та максимумом додатних елементів масиву. Вивести екстремальні елементи та знайдену різницю.

4 В веденому масиві цілих чисел знайти та вивести мінімум непарних елементів масиву та максимум парних елементів. Відповідні екстремальні елементи масиву зменшити вдвічі. Вивести оновлений масив.

5 В веденому масиві дійсних чисел знайти мінімальний серед елементів, значення яких належать діапазону $[-6 ; 4]$ та максимальний серед елементів, значення яких належать діапазону $[5 ; 14]$.

6 В веденому масиві дійсних чисел знайти перший максимум додатних елементів масиву. Вивести знайдений максимум та його номер. Всі від'ємні елементи масиву замінити на знайдений максимум або на задане користувачем число в разі відсутності додатних елементів. Вивести оновлений масив.

7 В веденому масиві цілих чисел знайти добуток максимуму непарних елементів масиву та мінімуму парних. Вивести значення екстремальних елементів масиву та знайдений добуток.

8 В веденому масиві дійсних чисел знайти перший мінімум серед від'ємних елементів, що розташовані на парних місцях. Вивести знайдений мінімум та його номер. Всі елементи масиву, що розташовані на непарних місцях замінити на знайдений мінімум або на введене користувачем число. Вивести оновлений масив.

9 В веденому масиві цілих чисел знайти мінімум парних елементів масиву та максимум непарних елементів. Вивести знайдені екстремуми. Всі додатні елементи масиву замінити на суму знайдених екстремумів. Вивести оновлений масив.

10 В веденому масиві цілих чисел знайти різницю між мінімумом непарних елементів масиву, розташованих на парних місцях та максимумом парних елементів масиву, розташованих на непарних місцях. Вивести екстремальні елементи та знайдену різницю.

11 В веденому масиві дійсних чисел знайти мінімум додатних елементів масиву, розташованих на парних місцях. Вивести знайдений мінімум. Всі від'ємні елементи масиву замінити на знайдений мінімум або введене користувачем число. Вивести оновлений масив.

12 В веденому масиві дійсних чисел знайти перший максимум серед від'ємних елементів, що розташовані на непарних місцях. Вивести знайдений

максимум та його номер. Всі додатні елементи масиву замінити на знайдений максимум або введене користувачем число. Вивести оновлений масив.

13 В веденому масиві дійсних чисел знайти суму максимуму від'ємних елементів та мінімуму додатних елементів масиву. Вивести екстремальні елементи та знайдену суму. Всі нульові елементи масиву замінити знайденою сумою. Вивести оновлений масив.

14 В веденому масиві дійсних чисел знайти та вивести перший максимум елементів масиву та його номер. Всі від'ємні елементи масиву замінити на знайдений максимум. Вивести оновлений масив.

15 В веденому масиві дійсних чисел знайти мінімальний серед елементів, значення яких належать діапазону $[6; 50]$ та максимальний серед елементів, значення яких належать діапазону $[-15; -4]$. У разі відсутності в масиві елементів із вказаних діапазонів знайти мінімум та максимум серед всіх елементів масиву.

16 В веденому масиві цілих чисел знайти добуток максимуму парних елементів масиву, розташованих на парних місцях та мінімуму непарних елементів масиву, розташованих на непарних місцях. Вивести екстремальні елементи та знайдений добуток.

17 В веденому масиві цілих чисел знайти та вивести максимум непарних елементів масиву. Всі додатні елементи масиву замінити на знайдений максимум або на введене користувачем число. Вивести оновлений масив.

18 В веденому масиві дійсних чисел знайти та вивести максимум від'ємних елементів. Додатні елементи масиву замінити на знайдений максимум або на введене користувачем число в разі відсутності в масиві від'ємних елементів. Вивести оновлений масив.

19 В веденому масиві цілих чисел знайти та вивести максимум парних елементів масиву та мінімум непарних елементів. Всі від'ємні елементи масиву замінити на різницю знайдених екстремумів. Вивести оновлений масив.

20 В веденому масиві дійсних чисел знайти мінімум серед елементів, значення яких лежать в діапазоні $[-18; -1]$ та максимум серед елементів, значення яких лежать в діапазоні $[1; 18]$. Вивести знайдені екстремуми. Всі нульові елементи масиву замінити різницею знайдених екстремумів або на введене користувачем число. Вивести оновлений масив.

21 В веденому масиві дійсних чисел знайти останній максимум серед від'ємних елементів, що розташовані на парних місцях. Вивести знайдений максимум та його номер. Всі додатні елементи масиву замінити на знайдений максимум або на введене користувачем число. Вивести оновлений масив.

22 В веденому масиві дійсних чисел знайти перший мінімум від'ємних елементів масиву. Вивести знайдений мінімум та його номер. Всі додатні елементи

масиву замінити на знайдений мінімум або на введене користувачем число. Вивести оновлений масив.

23 В веденому масиві дійсних чисел знайти різницю першого максимуму від'ємних елементів та останнього мінімуму додатних елементів масиву. Вивести екстремальні елементи, їх номери та знайдену різницю.

24 В веденому масиві дійсних чисел знайти та вивести останній мінімум елементів масиву та його номер. Всі додатні елементи масиву замінити на знайдений мінімум. Вивести оновлений масив.

25 В веденому масиві дійсних чисел знайти добуток між максимумом від'ємних елементів та мінімумом додатних елементів масиву. Вивести екстремальні елементи та знайдений добуток.

26 В веденому масиві цілих чисел знайти суму мінімуму непарних елементів масиву, розташованих на парних місцях та максимуму парних елементів масиву, розташованих на непарних місцях. Вивести екстремальні елементи та знайдену суму.

27 В веденому масиві цілих чисел знайти максимум парних елементів масиву та мінімум непарних елементів. Вивести знайдені екстремуми. Всі додатні елементи масиву замінити на суму знайдених екстремумів. Вивести оновлений масив.

28 В веденому масиві дійсних чисел знайти мінімальний серед елементів, значення яких належать діапазону $[-60; -5]$ та максимальний серед елементів, значення яких належать діапазону $[-1; 40]$. У разі відсутності в масиві елементів із вказаних діапазонів знайти мінімум та максимум серед всіх елементів масиву.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Як сформувати псевдодинамічний масив?
- 2 Який фактичний розмір псевдо динамічного масиву?
- 3 Чому масив називається псевдодинамічним?
- 4 Опишіть алгоритм пошуку максимального (мінімального) елемента масиву.
- 5 Як знайти перший максимальний (мінімальний) елемент масиву?
- 6 Як знайти останній максимальний (мінімальний) елемент масиву?
- 7 В якому випадку максимальний елемент масиву дорівнюватиме мініальному?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 9
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм сортування лінійних масивів даних

Мета: Придбати практичні навички з складання та налагодження програм реалізації наступних алгоритмів сортування лінійних масивів даних – обмінами, вставками, вибором..

Час 4 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

3 Алгоритми сортування

Сортування масивів полягає у розстановці елементів масиву у певному порядку – за зростанням або за спаданням. В залежності від способу порівняння і обміну елементів розрізняють різні типи алгоритмів сортування.

Основними алгоритмами сортування є:

- сортування масиву методом вибору;
- сортування обмінами («бульбашковий» метод сортування);
- сортування простими вставками.

3.1 Сортування масиву методом вибору.

Одномірний масив із n цілих чисел відсортуємо за зростанням значень елементів методом вибору. Алгоритм полягає в тому, що вибирається найменший елемент і міняється місцями з першим елементом масиву, потім розглядаються елементи масиву, починаючи із другого, і найменший з них міняється місцями із другим елементом, і так далі $n-1$ раз (при останньому проході циклу при необхідності міняються місцями передостанній й останній елементи масиву).

3.2 Сортування обмінами («бульбашковий» метод сортування)

Одномірний масив із n цілих чисел відсортуємо за зростанням значень елементів. Алгоритм полягає в тому, що при перегляді вхідного масиву попарно порівнюються сусідні елементи. Якщо порядок їхнього проходження не відповідає

заданому критерію впорядкованості, то елементи міняються місцями. В результаті одного такого перегляду при сортуванні за збільшенням елементів елемент з найбільшим значенням переміститься на останнє місце в масиві. При наступному проході на своє місце переміститься другий за величиною елемент і т.д. Для постановки на свої місця n елементів слід зробити $n-1$ проходів. При кожному черговому проході кількість елементів, що порівнюються, треба зменшувати на одиницю.

3.3 Сортування простими вставками

Одномірний масив $b[]$ із n цілих чисел відсортуємо за зростанням значень елементів методом вибору. Ідея алгоритму: для кожного елемента масиву $b[i]$ вважаємо, що попередні елементи вже відсортовані. Елемент $b[i]$ потрібно вставити на відповідне місце в вже відсортовану послідовність $b[0] \dots b[i-1]$. Це місце визначається послідовним порівнянням $b[i]$ з вже впорядкованими елементами i , якщо необхідно - елементи міняються місцями.

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 *Провести аналіз поставленої задачі.*

Загальна постановка завдання:

Дано лінійний масив з n елементів (n задає користувач). Обробити масив за алгоритмом згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- сформувати псевдодинамічний масив;
- введення значень елементів масиву проводити в окремому циклі;
- відсортувати лінійний масив методом, вказаним в Вашому варіанті;
- вивести в стовбець відсортований масив даних;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 *Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.*

3 *Розробити та налагодити програму рішення задачі мовою програмування C/C++.*

4 *Оформити звіт з лабораторної роботи.*

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Вхідні дані	Завдання на обробку
1	A(n) – масив дійсних чисел	Відсортувати масив за зростанням елементів «бульбашковим» методом (сортування обмінами).
2	B(n) – масив дійсних чисел	Відсортувати масив за спаданням елементів простими вставками.
3	V(n) – масив дійсних чисел	Відсортувати масив за неспаданням елементів методом вибору.
4	M(n) – масив дійсних чисел	Відсортувати масив за спаданням елементів «бульбашковим» методом (сортування обмінами).
5	A(n) – масив цілих чисел	Відсортувати масив за незростанням елементів простими вставками.
6	H(n) – масив цілих чисел	Відсортувати масив за зростанням елементів методом вибору.
7	P(n) – масив цілих чисел	Відсортувати масив за незростанням елементів «бульбашковим» методом (сортування обмінами).
8	L(n) – масив цілих чисел	Відсортувати масив за зростанням елементів простими вставками.
9	T(n) – масив дійсних чисел	Відсортувати масив за спаданням елементів методом вибору.
10	N(n) – масив дійсних чисел	Відсортувати масив за неспаданням елементів «бульбашковим» методом (сортування обмінами).
11	A(n) – масив дійсних чисел	Відсортувати масив за неспаданням елементів простими вставками.
12	X(n) – масив дійсних чисел	Відсортувати масив за незростанням елементів методом вибору.

Варіант	Вхідні дані	Завдання на обробку
13	$K(n)$ – масив дійсних чисел	Відсортувати масив за зростанням елементів «бульбашковим» методом (сортування обмінами).
14	$D(n)$ – масив дійсних чисел	Відсортувати масив за спаданням елементів простими вставками.
15	$R(n)$ – масив дійсних чисел	Відсортувати масив за неспаданням елементів методом вибору.
16	$S(n)$ – масив дійсних чисел	Відсортувати масив за спаданням елементів «бульбашковим» методом (сортування обмінами).
17	$H(n)$ – масив дійсних чисел	Відсортувати масив за незростанням елементів простими вставками.
18	$M(n)$ – масив дійсних чисел	Відсортувати масив за зростанням елементів методом вибору.
19	$R(n)$ – масив дійсних чисел	Відсортувати масив за незростанням елементів «бульбашковим» методом (сортування обмінами).
20	$F(n)$ – масив дійсних чисел	Відсортувати масив за зростанням елементів простими вставками.
21	$Z(n)$ – масив цілих чисел	Відсортувати масив за спаданням елементів методом вибору.
22	$C(n)$ – масив цілих чисел	Відсортувати масив за неспаданням елементів «бульбашковим» методом (сортування обмінами).
23	$V(n)$ – масив цілих чисел	Відсортувати масив за неспаданням елементів простими вставками.
24	$M(n)$ – масив цілих чисел	Відсортувати масив за незростанням елементів методом вибору.
25	$W(n)$ – масив цілих чисел	Відсортувати масив за зростанням елементів «бульбашковим» методом (сортування обмінами).
26	$Y(n)$ – масив цілих чисел	Відсортувати масив за спаданням елементів простими вставками.
27	$P(n)$ – масив цілих чисел	Відсортувати масив за неспаданням елементів методом вибору.

Варіант	Вхідні дані	Завдання на обробку
28	A(n) – масив цілих чисел	Відсортувати масив за спаданням елементів «бульбашковим» методом (сортування обмінами).

КОНТРОЛЬНІ ПИТАННЯ

- 1 Яка ідея полягає в основі алгоритму сортування обмінами?
- 2 Опишіть алгоритм сортування методом вибору?
- 3 Опишіть алгоритм сортування вставками?
- 4 Скільки проходів по масиву відбувається при сортуванні обмінами? вибором? вставками?
- 5 На вхід подається відсортований масив. Як вдосконалити метод обмінами так, щоб сортування припинялось вже після першого проходу по масиву?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 10
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм обробки матриць.

Мета: Придбати наступні практичні навички при складанні та налагодженні програм обробки матриць:

- опис матриці;
- формування псевдодинамічної матриці;
- використання вкладених циклів для введення/виведення та обробки матриці;
- форматоване виведення матриці.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Двовимірні масиви – матриці

При визначенні двовимірних масивів (матриць) задається тип компонентів масиву, ім'я масиву, а потім, в окремих квадратних дужках, число елементів у першому "вимірі" та другому. Наприклад, двовимірний масив цілих чисел із двох рядків і трьох стовпців можна визначити так:

```
int A[2][3];          // 2 рядка масивів по 3 елементи в кожному
```

Матриці можна ініціалізувати за допомогою синтаксису, аналогічного ініціалізації одномірних масивів: початкові значення перераховані в блоці з роздільниками-комами:

```
int m[2][3]={ 10, 20, 30, 40, 50, 60 };
```

Тут перші три значення попадають у перший рядок матриці, а останні три - у другий рядок. Для більших масивів можна вказати групу значень, що належать до одного рядка, за допомогою дужок, які визначають область дії. Ініціалізатори розділяються комами, це допоможе супроводжуючому додаток програмістові ідентифікувати дані в кожному рядку:

```
int m[2][3]={ { 10, 20 , 30 }, { 40, 50 , 60 } };
```

Початкових значень, як й в одномірних масивах, може бути менше, ніж число елементів масиву. Іншим елементам буде привласнюватися нульове значення. Наприклад:

```
int m[2][3]={ { 10, 20 }, { 30, 40 } };
```

Це еквівалентно наступному явному визначенню, де перші три значення попадають у перший рядок, а останні - у другий:

```
int m[2][3]={ 10, 20, 0, 30, 40, 0 };
```

Подібно одномірним масивам, число початкових значень не повинне перевищувати числа елементів у рядку:

```
int m[2][3]={ { 10, 20, 30, 40 }, { 50, 60 } };    //  
помилка
```

Метод визначення розміру масиву шляхом специфікації початкових значень може використатися й для двовимірних масивів, але тільки частково. Дозволяється опускати число рядків, але потрібно задавати число стовпців. Компілятор буде підраховувати число початкових значень і визначати число рядків.

```
int m[ ][3]={ { 10, 20, 30 }, { 40, 50, 60 } };
```

Незалежно від того, указується число рядків чи ні, не можна опускати число стовпців, це помилка:

```
int m[2][]={ { 10, 20, 30 }, { 40, 50, 60 } }    //  
помилка
```

Теоретично компілятор може підраховувати групи рядків і розуміти структуру матриці, однак він цього не робить. Можливо, краще задавати розмірність масивів явно.

Для доступу до елементів багатомірного масиву потрібно кілька індексів – по одному на кожен розмірність. Аналогічно одномірним масивам, кожен індекс представляє зсув елемента, а отже, починається з 0 і закінчується значенням, на одиницю меншим числа елементів масиву. Наприклад, перший елемент другого рядка матриці `m[ ][ ]` позначається як `m[1][0]`.

2 Обробка двовимірних масивів

При переборі елементів матриці варто використати вкладені цикли. У вкладених циклах внутрішній цикл завершує всі свої операції спочатку для однієї ітерації зовнішнього циклу, потім для наступної ітерації зовнішнього циклу й т.д.

2.1 Обхід матриці по рядках

При обході матриці по рядках другий індекс елементу матриці (номер стовпця) змінюється частіше. А тому зовнішній цикл організують по першому індексу (номеру рядка), а внутрішній цикл – по другому індексу (номеру стовпця). Наведемо приклад фрагменту програми для обчислення середнього арифметичного значення в кожному з рядків заданої матриці цілих чисел $B(5,6)$:

```
. . .
int b[5][6]; // Опис заданої матриці
float sa; // Середнє арифметичне рядка матриці
. . .
for (int i=0; i<5; i++) // Зовнішній цикл по номеру рядка
{
    sa=0;
    for (int j=0; j<6; j++) // Внутрішній цикл по номеру стовпця
        sa+=b[i][j];
    cout<<endl<<"Среднее арифметическое "<<i+1<<"-й строки равно
"<<sa/6;
}
. . .
```

2.2 Обхід матриці по стовпцях

При обході матриці по стовпцях перший індекс елементу матриці (номер рядка) змінюється частіше. А тому зовнішній цикл організують по другому індексу (номеру стовпця), а внутрішній цикл – по першому індексу (номеру рядка). Наведемо приклад фрагменту програми для обчислення кількості додатних елементів в кожному з стовпців заданої матриці дійсних чисел $M(10,12)$:

```
. . .
float m[10][12]; // Опис заданої матриці
int k; // Кількість додатних елементів в рядку матриці
. . .
for (int j=0; j<12; j++) // Зовнішній цикл по номеру стовпця
{
    k=0;
    for (int i=0; i<10; i++) // Внутрішній цикл по номеру рядка
        if ( m[i][j]>0) k++;
    if (k==0)
```



```

        cout<<endl<<"В столбце "<<j+1<<" нет положительных
элементов";
    else
        cout<<endl<<"Количество положительных в "<<j+1<<"-м столбце равно
"<<k;
    }
    . . .

```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Скласти та налагодити програму введення матриці $A(N,M)$ (N та M вводяться користувачем) та обробки її за алгоритмом згідно Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- Формування псевдодинамічної матриці;
- введення значень елементів матриці проводити в окремому циклі;
- вивести введену матрицю;
- при обробці матриці враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

1. Дано матрицю дійсних чисел $A(N,M)$. Обчислити середнє арифметичне додатних елементів у кожному із стовпчиків.
2. Дано матрицю цілих чисел $A(N,M)$. Обчислити кількість додатних непарних елементів у кожному з рядків.
3. Дано матрицю цілих чисел $A(N,M)$. Обчислити кількість від'ємних парних елементів у кожному із стовпчиків.
4. Дано матрицю дійсних чисел $A(N,M)$. Обчислити середнє арифметичне від'ємних елементів у кожному з рядків.
5. Дано матрицю цілих чисел $A(N,M)$. Обчислити кількість додатних парних елементів у кожному із стовпчиків.
6. Дано матрицю цілих чисел $A(N,M)$. Обчислити середнє арифметичне від'ємних непарних елементів у кожному з рядків.
7. Дано матрицю цілих чисел $A(N,M)$. Обчислити суму додатних непарних елементів у кожному із стовпчиків.
8. Дано матрицю дійсних чисел $A(N,M)$. Обчислити добуток додатних елементів у кожному з рядків.
9. Дано матрицю дійсних чисел $A(N,M)$. Обчислити кількість від'ємних елементів у кожному із стовпчиків.
10. Дано матрицю цілих чисел $A(N,M)$. Обчислити суму додатних парних елементів у кожному з рядків.
11. Дано матрицю дійсних чисел $A(N,M)$. Обчислити кількість від'ємних парних елементів у кожному із стовпчиків.
12. Дано матрицю цілих чисел $A(N,M)$. Обчислити суму від'ємних парних елементів у кожному з рядків.
13. Дано матрицю цілих чисел $A(N,M)$. Обчислити добуток непарних елементів у кожному із стовпчиків.
14. Дано матрицю дійсних чисел $A(N,M)$. Обчислити кількість нульових елементів у кожному з рядків.
15. Дано матрицю дійсних чисел $A(N,M)$. Обчислити добуток додатних елементів у кожному із стовпчиків.
16. Дано матрицю дійсних чисел $A(N,M)$. Обчислити суму ненульових елементів у кожному з рядків.

17. Дано матрицю цілих чисел $A(N,M)$. Обчислити суму додатних непарних елементів у кожному із стовпчиків.

18. Дано матрицю дійсних чисел $A(N,M)$. Обчислити кількість від'ємних елементів у кожному з рядків.

19. Дано матрицю дійсних чисел $A(N,M)$. Обчислити кількість ненульових елементів у кожному із стовпчиків.

20. Дано матрицю дійсних чисел $A(N,M)$. Обчислити суму від'ємних елементів у кожному з рядків.

21. Дано матрицю дійсних чисел $A(N,M)$. Обчислити добуток ненульових елементів у кожному із стовпчиків.

22. Дано матрицю цілих чисел $A(N,M)$. Обчислити кількість додатних парних елементів у кожному з рядків.

23. Дано матрицю цілих чисел $A(N,M)$. Обчислити середнє арифметичне непарних елементів у кожному із стовпчиків.

24. Дано матрицю дійсних чисел $A(N,M)$. Обчислити кількість нульових елементів у кожному з рядків.

25. Дано матрицю дійсних чисел $A(N,M)$. Обчислити середнє арифметичне від'ємних елементів у кожному із стовпчиків.

26. Дано матрицю цілих чисел $A(N,M)$. Обчислити кількість від'ємних непарних елементів у кожному з рядків.

27. Дано матрицю дійсних чисел $A(N,M)$. Обчислити суму додатних елементів у кожному із стовпчиків.

28. Дано матрицю цілих чисел $A(N,M)$. Обчислити добуток від'ємних парних елементів у кожному з рядків.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Що називається багатовимірним масивом?
- 2 Наведіть загальну форму оголошення багатовимірного масиву.
- 3 З якого числа починається нумерація індексів багатовимірного масиву?
- 4 Як розташовується в пам'яті багатовимірний масив?
- 5 Наведіть загальну форму оголошення двовимірного масиву.
- 6 Наведіть приклад ініціалізації багатовимірного масиву.
- 7 Скільки і яку назву мають індекси, необхідні для звернення до елемента матриці?
- 8 Як визначити загальний розмір пам'яті, який займе матриця?
- 9 Які ініціалізатори використовує компілятор у випадку, коли у підгрупах ініціалізаторів не вистає членів підгруп?

- 10 Які типи циклів і в якому порядку зазвичай використовують для виконання будь-яких дій над матрицями?

Методичні рекомендації до виконання лабораторної роботи № 11

з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм з використанням масивів структур

Мета: Придбати практичні навички конструювання структур даних та реалізації програм обробки масивів структур.

Час 4 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Оголошення структури, доступ до полів структури. Робота з масивами структур

Зберігати дані в масивах не завжди зручно: масиви можуть містити елементи тільки одного типу, і якщо дані будуть різного типу, необхідно їх розкидати по різних масивах. Наприклад, програма повинна працювати з базою даних про співробітників фірми. Спрощено, у такій базі повинне бути хоча б прізвище, код, рік народження й зарплата. Прізвище - це символьний рядок, рік народження й код - цілі числа, зарплата - речовинне число. Якщо користуватися масивами для роботи з такою базою, прийде зберігати прізвища в масиві символьних рядків, роки народження й коди - у масивах `int`, а зарплату - у дійсному масиві. Утомливо.

Для роботи з такими конструкціями в C використовуються структури. Структура - це набір елементів, можливо різного типу. Структури мають імена, і елементи цієї структури (поля структури) мають імена. Синтаксис опису структури:

```
struct [имя_типа] {  
    тип_1 элемент_1;  
    тип_2 элемент_2;  
    ...  
    тип_n элемент_n;  
} [список_описателей];
```

Наприклад:

```
struct Worker {    //опис нового типу Worker
```

```
char fio[30];
int date, code;
double zarplata;
}; // опис закінчується крапкою з комою
```

Ім'я описаного типу можна використати надалі поряд зі стандартними типами, наприклад можна визначити масив типу Worker:

```
Worker stuff [100];
```

Елементи структури називаються полями структури й можуть мати будь-який тип, крім типу цієї ж структури. Якщо відсутнє ім'я типу, повинен бути зазначений список змінних або масивів. У цьому випадку опис структури служить визначенням елементів цього списку:

Визначення структури та масиву структур:

```
struct {
    char fio[30];
    int date, code;
    double zarplata;
} emp, stuff [100];
```

Для ініціалізації структури значення її елементів перераховують у фігурних дужках у порядку їхнього опису:

```
struct {
    char fio[30];
    int date, code;
    double zarplata;
} worker = {"Страусенко", 31, 215, 3400.55};
```

При ініціалізації масивів структур варто брати у фігурні дужки кожен елемент масиву (з огляду на, що багатомірний масив - це масив масивів):

```
struct coordinata{
    int x, y;
} coord[2][3] = {
    {{1, 1}, {3, 2}, {-1,4}}, //рядок 1, тобто масив coord[0]
    {{-1, 0}, {1, -2}, {1,0}}, //рядок 2, тобто масив coord[1]
};
```

Для змінних того самого структурного типу визначена операція присвоювання, при цьому відбувається заелементне копіювання. Розмір структури не обов'язково дорівнює сумі розмірів її елементів.

Доступ до полів структури здійснюється за допомогою операції вибору . (крапка) при звертанні до поля через ім'я структури, наприклад:

```
Worker worker, stuff [100];  
...  
worker.fio = "Страусенко";  
stuff [8].code = 215;
```

Якщо елементом структури є інша структура, то доступ до її елементів виконується через дві операції вибору:

```
struct A {int a; double x;}  
  struct B {A a; double x;} x[2];  
x[0].a.a = 1;  
x[1].x = 0.1;
```

Поля різних структур можуть мати однакові імена, оскільки в них різна область видимості.

От приклад опису структури, оголошення й доступу до її елементів:

```
struct NAME{           //описуємо структуру NAME  
//вона має три елементи - три символічних масиви  
  char familiya[30], imya[30], otchestvo[30];  
}  
  
struct sluj{           //описуємо структуру sluj  
//до неї входить структура sluj_name типу NAME і змінна year типу int  
  NAME sluj_name;  
  int year;  
}  
  
void main()  
{  
  sluj Masha;          //описуємо змінну-структуру Masha типу sluj  
//працюємо з елементами структури sluj_name, вкладеної в Masha:  
  Masha.sluj_name.familiya="Іванова";  
  Masha.sluj_name.imya="Марья";  
  Masha.sluj_name.otchestvo="Степанівна";  
  
  // працюємо з елементом структури Masha  
  Masha.year=1917;  
  
  sluj Petya; //Повідомляємо структуру Petya типу sluj
```

Petya.year=1943; //рік народження Петі ніяк не впливає на рік народження
Маші

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Створити псевдодинамічний масив структур даних. Обробити його за алгоритмом згідно Вашого варіанта. Програма повинна задовольняти наступним вимогам:

- повинні бути реалізовані наступні функції обробки масиву структур:
 - 1 введення значень елементів масиву;
 - 2 виведення масиву в загальноприйнятому вигляді;
 - 3 функції обробки масиву структур за завданнями згідно з індивідуальними завданнями Вашого варіанту.
- організувати меню для вибору користувачем варіанту виконання вказаних вище функцій. Забезпечити повернення в меню програми після виконання чергового пункту меню. Вихід із програми здійснювати по відповідному пункту меню: «Выход из программы»;
- при обробці масиву враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування Сі/С++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування С/С++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

№	Умова задачі	Індивідуальні завдання
1	Скласти програму, що містить відомості про вогнепальну зброю МВС області. Структура повинна складатися з полів: • модель; • табельний номер; • калібр; • маса; • швидкострільність; • ПІБ власника.	Видати наступну інформацію: 1. ПІБ власників, що мають найбільшу в області кількість одиниць зброї калібру X із вказанням їх характеристик; 2. Перелік зброї масою не менш за M калібру K. X, M та K – вводити з клавіатури.
2	Скласти програму, що містить відомості про асортимент продержаних авто у автопарку «Власник». Структура повинна складатися з полів: • Марку авто; • Колір; • Потужність; • Рік випуску; • Пробіг; • ПІБ власника.	Видати наступну інформацію: 1. Переліки найстаріших та найновіших авто марки M. 2. Марки авто, що мають потужність не менш за X останніх 5-х років випуску із вказанням їх власників. X, M – вводити з клавіатури.
3	Скласти програму, що містить відомості про соціально незахищені сім'ї міста. Структура повинна складатися з полів: • Категорія – малозабезпечений, чорнобилець, інвалід(група), багатодітний; • Кількість членів у сім'ї; • Район мешкання;	Видати наступну інформацію: 1. Перелік сімей з найбільшою кількістю мешканців району X; 2. Перелік сімей з помешканням виду M категорії K. X, M , K – вводити з клавіатури

№	Умова задачі	Індивідуальні завдання
	Вид помешкання – особистий дім, багатоповерхівка, барак, без помешкання.	
4	Скласти програму, що містить відомості про ДТП області. Структура повинна складатися з полів: - Район ДТП; - Дата; - ПІБ власника авто, що скоїв ДТП; - Кількість постраждалих.	Видати наступну інформацію: - Кількість постраждалих у ДТП в районі Х за у період часу від А до В(дата). - ПІБ винуватців ДТП, де постраждалих найбільша кількість та ДТП, де не було постраждалих. Х, М , К – вводити з клавіатури
5	Скласти програму, що містить відомості про проходження диспансеризації у коледжі. Структура повинна складатися з полів: - ПІБ студента; - Група; - ПІБ керівника групи; - Рік народження; - Стать; - Ознака – чи пройдено флюорографію; - Ознака – чи пройдено стоматолога; - Ознака – чи пройдено терапевта;	Видати наступну інформацію: Перелік груп, де відсоток студентів, що не пройшли чи пройшли не повністю диспансеризацію - найбільший; Перелік студентів статі К, що не пройшли вид огляду Х із вказанням групи. Х, К – вводити з клавіатури
6	Скласти програму, що містить відомості про асортимент канцтоварів у магазині «Олівець». Структура повинна складатися з полів: назву виробу;	Видати наступну інформацію: Ціни 2-х найдорожчих виробів країни-виробника Y і їх назви;

№	Умова задачі	Індивідуальні завдання
	▪ ціну; ▪ країну- виробника; ▪ кількість одиниць; ▪ модель.	2) Перелік виробів цінової категорії від А до В у кількості не менш за Х одиниць. Х, А, В та Y – вводити з клавіатури.
7	Скласти програму, що містить відомості про здачу студентами 3 курсу відділення 1 сесії. Структура повинна складатися з полів: ▪ Група; ▪ ПІБ студента; ▪ Відмітки по 2 іспитам; ▪ Відмітки по 3 залікам; ▪ Ознака участі в суспільній праці (активна \ неактивна); ▪ База навчання (бюджет/ контракт)	Видати відомість нарахування студентам- бюджетникам групи Х стипендії. Студент, що отримав всі відмітки «5» і активно брав участь в суспільній праці, зараховується на підвищену стипендію - доплата 50%, неактивно брав участь – доплата 25%. Студенти, що отримали «4» і «5», зараховуються на звичайну стипендію. Студент, що отримав одну оцінку «3», але активно займався суспільною працею, також зараховується на звичайну стипендію, інакше зачислення не буде. Х та розмір звичайної стипендії – вводити з клавіатури.
8	Скласти програму, що містить відомості про здачу студентами коледжу сесії. Структура повинна складатися з полів: ▪ індекс групи; ▪ ПІБ студента; ▪ вік; ▪ стать; ▪ Відмітки по п'яти залікам і іспитам.	Видати наступну інформацію: 1) ПІБ невстигаючих студентів з вказаним індексом групи статі Y та кількості заборгованостей; 2) Середній бал, отриманий кожним студентом групи Х, і всієї групи в цілому. Х, Y, Z – вводити з клавіатури.

№	Умова задачі	Індивідуальні завдання
9	Скласти програму, що містить відомості про наявність білетів і рейсах Аерофлоту м. Дн-ська. Структура повинна складатися з полів: ■ номер рейса; ■ пункт призначення; ■ час вилита; ■ час прибуття; ■ признак – є топливна дозаправка чи ні; ■ кількість вільних місць в салоні.	Видати наступну інформацію: 1) Час відправки літака в місто X; 2) Кількість вільних місць на рейс в місто X з часом відправки Y. 3) відомості про літаки, що без дозаправки знаходяться у повітрі більш за Z годин. X,Y – вводити з клавіатури.
10	Скласти програму, яка містить відомості про наявність взуття в магазині фірми «Підбор». Структура повинна складатися з полів: ■ артикул; ■ найменування; ■ кількість; ■ колір; ■ виробник; ■ матеріал; ■ ціна однієї пари.	Видати наступну інформацію: 1) Асортиментний список взуття із матеріалу E з вказівкою наявності кількості пар кожної моделі. 2) Інформацію про взуття найменування Y кольору Z. Y,Z,E – вводити з клавіатури.
11	Скласти програму, яка містить відомості про п'ять дисциплін, які на вибір бажає слухати студент. Структура повинна складатися з полів: ■ ПІБ студента; ■ Індекс групи; ■ 2 дисципліни; ■ стаття; ■ середній бал успішності;	Надрукувати список студентів статі Y, бажаючих прослухати дисципліну X. Якщо кількість бажаючих перевищує 8 чоловік, то відібрати студентів, що мають більш високий середній бал успішності. X, Y – вводити з клавіатури.

№	Умова задачі	Індивідуальні завдання
12	Скласти програму, яка містить відомості про відправку поїздів дальнього проходження з Дніпропетровського вокзалу. Структура повинна складатися з полів: ▪ номер поїзда; ▪ час відправки; ▪ станція призначення; ▪ ПІБ начальника составу; ▪ Кількість вагонів; ▪ наявність білетів.	Видати наступну інформацію: 1) Перелік. поїздів з номерами від А до В, де кількість вагонів мінімальна; 2) Наявність білетів на поїзд з номером Z. А, В, Z – вводити з клавіатури.
13	Скласти програму, яка містить відомості про співробітників коледжу. Структура повинна складатися з полів: ▪ ПІБ робітника; ▪ Назва відділення; ▪ Рік народження; ▪ Стаж роботи; ▪ Посада; ▪ Стать; ▪ Оклад.	Видати наступну інформацію: 1) Список співробітників статі С пенсійного віку на сьогоднішній день. 2) Середній стаж працюючих в відділенні Х. 3) Список співробітників з найменшим окладом; Х, С – вводити з клавіатури.
14	Скласти програму, що зберігає дані про товари на складі. «Забудовник». Структура повинна складатися з полів: ▪ назву товару; ▪ кількість товару; ▪ дату виготовлення; ▪ дату прийому товару; ▪ термін зберігання. ▪ умови зберігання – температура, вологість.	Видати наступну інформацію: 1) Вивести дані про товар Х та перелік тих товарів, кількість яких перевищує N з терміном зберігання не менш за М. 2) Списки товарів (окремо), які повинні зберігатися при температурі від Н до К та при вологості не більш за 60 %. Х, N, М, Н, К – ввести з клавіатури.

№	Умова задачі	Індивідуальні завдання
15	Скласти програму, що зберігає дані про препарати аптек у м. Дн-ськ. Структура повинна складатися з полів: ▪ назва препарату; ▪ наявність препарату ; ▪ № аптеки; ▪ Виробник; ▪ ціна; ▪ Ознака: відпускається препарат за рецептом чи ні.	1) Вивести перелік препаратів за найвищою ціною в аптеці №N, що відпускаються без рецептів; 2) Вивести список препаратів виробника X, ціна яких знаходиться у межах A-B. X, N, A та B ввести з клавіатури.
16	Скласти програму, що зберігає дані про матеріальні цінності в аудиторіях коледжу. Структура повинна складатися з полів: • назва матеріальної цінності; • інвентарний номер; • номер аудиторії • ознака - полягає списанню чи ні. • ПІБ відповідального за матеріальні цінності цієї аудиторії	1. Вивести список усіх матеріальних цінностей, що полягають списанню в аудиторіях з номерами з номерами від N до M. 2. Вивести перелік усіх аудиторій, де є в наявності матеріальна цінність з назвою X з їх інвентарними номерами. 3. Вивести також список усіх матеріальні цінності відповідального А. N, M, X та A ввести з клавіатури.
17	Скласти програму, що веде облік товарів кіосків міста. Структура повинна складатися з полів • назва товару; • номери кіосків де товар в наявності (не більше 3); • ціна; • виробник товару. • гатунок товару. • Наявність ГМО.	1) Вивести перелік усіх кіосків, де в наявності товар X із вмістом ГМО. 2) Вивести перелік товарів гатунку Y за найвищою ціною у кількості не менш за N. X, Y,N та A ввести з клавіатури.

№	Умова задачі	Індивідуальні завдання
18	Скласти програму, що веде облік відвідувань занять студентів групи. Структура повинна складатися з полів: ■ ПІБ студента; ■ Кількість пропусків по дням тижня; ■ Відмітка, про причину пропусків (уважна чи неуважна) по дням тижня. ■ Група ■ Відділення.	1) Вивести перелік усіх студентів, що мають кількість неповажних пропусків занять у кількості від N до M годин, вище 40 годин. 2) Вивести також перелік груп відділення X, в яких найвища кількість прогулів. X, N та M ввести з клавіатури.
19	Скласти програму, що зберігає дані про товари на складі електротоварів «Мегамакс». Структура повинна складатися з полів: ■ назву товару; ■ кількість товару; ■ дату прийому товару; ■ потужність товару..	1) Вивести перелік товарів, прийнятих у день A з потужністю не менш за B Ватт. 2) Вивести також дані про товари, що мають найменшу та найбільшу кількість за цінами від N до M. M, N, A та B – ввести з клавіатури.
20	Скласти програму, що веде облік товарів продуктових магазинів. Структура повинна складатися з полів ■ назва товару; ■ номер магазину; ■ ціна; ■ термін зберігання. ■ гатунок товару.	1) Вести дані про товари з найнижчим терміном зберігання 2) Вивести перелік магазинів, де є в наявності товар з назвою A гатунку B. X, A та B – ввести з клавіатури.
21	Скласти програму, що зберігає дані про продаж охолоджуючих напоїв у літні місяці по районах міста. Структура повинна складатися з полів: ■ Район;	1) Обчислити відсоткове відношення продаж району X відносно загальної суми по всіх районах взагалі.

№	Умова задачі	Індивідуальні завдання
	▪ Місяць; ▪ Загальну суму з продажу цього місяця. ▪ Загальну суму неліквіду цього місяця.	3) Вивести перелік районів з найменшою загальною сумою неліквіду району А за усі місяці. Х, А – ввести з клавіатури.
22	Скласти програму обліку заборгованостей за електроенергію по районах міста за роки з 2008 по 2010. Структура повинна складатися з полів: ▪ Район міста; ▪ Суму заборгованості по трьом останнім рокам окремо (2008, 2009, 2010).	1) Вивести перелік районів, у яких заборгованості з кожним роком зменшувалися. 2) Обчислити також рік, де заборгованість по району Х була найбільшою. Х – ввести з клавіатури.
23	Скласти програму, що містить дані про лікарів та пацієнтів міської лікарні «Швидкої допомоги». Структура повинна складатися з полів: • ПІБ пацієнта; • вік; • відділення (хірургія, гастро, шкіряне, терапевтичне); • ПІБ лікаря, • Діагноз хворого.	Видати наступну інформацію: 1. Діагнози пацієнтів віком старіше за 50 років лікаря Х. 2. Перелік усіх пацієнтів відділення У з діагнозом Z. Х, У, Z вводити з клавіатури.
24	Скласти програму, що містить дані про обслуговування периферійних пристроїв фірмою "Іонік". Структура повинна складатися з полів: • назва підприємства, що обслуговується; • модель пристрою, який обслуговується; • прізвище та ініціали майстра;	Видати наступну інформацію: 1. Прізвище та ініціали майстра, який обслуговував найбільшу кількість пристроїв моделі Х. 2. Кількість усіх виявлених неполадок за вказаним проміжком часу від У до Z. Х, У, Z вводити з клавіатури.

№	Умова задачі	Індивідуальні завдання
	• назву виявленої неполадки; • вартість усунення неполадки; • дата виконання.	
25	Скласти програму, що містить дані про банківські вклади. Структура повинна складатися з полів: • ПІБ власника вкладу; • сума вкладу; • річний % нарахувань; • сума нарахованих процентів; • дата створення вкладу; • дата повернення вкладу.	Видати наступні інформацію: 1. Видати інформацію про загальну суму всіх нарахованих процентів закладами особи Х. 2. Знайти та вивести на екран найбільший річний % нарахувань.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Дати визначення структури як типу даних.
- 2 Які формати визначення структурного шаблону ви знаєте?
- 3 Як здійснюється ініціалізація структур?
- 4 Як виконується операція привласнення для структур?
- 5 Як здійснюється доступ до елементів структури?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 12
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм обробки динамічних матриць з використанням вказівників

Мета: Придбати практичні навички роботи з Непар – пам'яттю при складанні та налагодженні програм створення та обробки динамічних матриць з використанням вказівників.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Динамічні масиви

Масиви, які розміщуються в динамічній пам'яті, називаються динамічними. Створюються динамічні масиви за допомогою операції `new`, при цьому необхідно вказати тип і розмірність:

```
int n = 100;    //змінна n для розмірності масиву
float *p = new float [n];
```

У цьому рядку створюється змінна-показчик на `float`, у динамічній пам'яті виділяється безперервна область, достатня для розміщення 100 елементів дійсного типу, і адреса її початку записується в показчик `p`. Динамічні масиви не можна при створенні ініціалізувати, і вони не обнуляються. Перевага динамічних масивів полягає в тому, що розмірність може бути змінної, тобто обсяг пам'яті, виділюваної під масив, визначається на етапі виконання програми. Доступ до елементів динамічного масиву здійснюється точно так само, як і до статичних. Наприклад, до 5-му елемента наведеного вище масиву можна звернутися як `p[5]` або `*(p+5)`. Пам'ять, зарезервована під динамічний масив за допомогою `new[]`, повинна звільнятися оператором `delete[]`:

```
delete [] p;
```

2 Багатомірні масиви

Масив, елементи якого самі є масивами, називається двовимірним масивом. Двовимірний масив можна собі представити як таблицю, що має рядки й стовпчики. Такий масив варто визначати із двома індексами, один із яких визначає кількість рядків таблиці, а другий установлює кількість стовпчиків. Багатомірні масиви задаються вказівкою кожного виміру у квадратних дужках, наприклад, оператор

```
int matr [6][8];
```

задає опис двовимірного масиву з 6 рядків 8 стовпців. У пам'яті такий масив розташовується в послідовних осередках порядково. Багатомірні масиви змінюються так, що при переході до наступного елемента швидше всього змінюється останній індекс. Для доступу до елемента двовимірного масиву вказуються всі його індекси, тобто номер рядка й номер стовпця усередині масиву, наприклад:

```
matr[3][6], matr[i][j].
```

При ініціалізації багатомірного масиву він представляється або як масив масивів, при цьому кожен масив поміщається у свої фігурні дужки (у цьому випадку розмірність при описі можна не вказувати), або задається загальний список елементів у тім порядку, у якому елементи розташовуються в пам'яті:

```
//масив представляється як масив масивів  
int mass2 [][] = { {1,1}, {0,2}, {1,0} };  
//задається загальний список елементів  
int mass2 [3][2] = {1, 1, 0, 2, 1, 0};
```

При роботі з багатомірними масивами використовуються вкладені цикли.

Для створення динамічного багатомірного масиву необхідно вказати в операції new всі його розмірності (сама ліва розмірність може бути змінної), наприклад:

```
int nstr = 5;  
int **m=(int**) new [nstr][10];
```

Більше універсальний і безпечний спосіб виділення пам'яті під двовимірний масив, коли обидві його розмірності задаються на етапі виконання програми, наступний:

```
int nstr, nstb;
cout<<" Уведіть кількість рядків і стовпців :";
cin>>nstr>>nstb;
int **a= new int *[nstr];      //1
for ( int i=0; i<nstr; i++)    //2
a[i] = new int [nstb];        //3
```

У операторі 1 оголошується змінна типу "показчик на показчик на int" і виділяється пам'ять під масив показчиків на рядки масиву (кількість рядків - nstr). В операторі 2 організується цикл для виділення пам'яті під кожен рядок масиву. В операторі 3 кожному елементу масиву показчиків на рядки привласнюється адреса початку ділянки пам'яті, виділеного під рядок двовимірного масиву. Кожен рядок складається з nstb елементів типу int.

Зауваження: якщо при описі змінної використовуються показчик й [] (масив), те змінна інтерпретується як масив показчиків, а не показчик на масив: int *p[10] - масив з 10 показчиків на int. Звільнення пам'яті з-під масиву з будь-якою кількістю вимірів виконується за допомогою операції delete []. Показчик на константу видалити не можна.

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Розмірність матриці (кількість рядків та стовпчиків) користувач вводить з клавіатури. Створити відповідну динамічну матрицю. Обробити її за алгоритмом згідно Вашого варіанта. Програма повинна задовольняти наступним вимогам:

- введення значень елементів матриці проводити в окремому циклі з відповідними коментарями;

- вивести введену матрицю в загальноприйнятому вигляді;
- при обробці матриці враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 *Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.*

3 *Розробити та налагодити програму рішення задачі мовою програмування C/C++.*

4 *Оформити звіт з лабораторної роботи.*

Звіт повинен містити наступні розділи:

1 Постановка задачі.

2 Блок-схема алгоритму рішення задачі.

3 Текст програми мовою програмування C/C++.

4 Копії вікон виконання програми.

Розробити тестові набори вхідних даних, що демонструють роботу програми по всіх можливих гілках розгалуження та помістити в звіт копії вікон виконання програми для кожного тестового набору.

5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

1. Знайти мінімальний та обчислити кількість нульових елементів у кожному з рядків матриці.
2. Знайти максимальний та обчислити середнє арифметичне від'ємних елементів у кожному із стовпчиків матриці.
3. Знайти максимальний та обчислити кількість від'ємних непарних елементів у кожному з рядків матриці.

4. Знайти максимальний та обчислити суму додатних елементів у кожному із стовпчиків.

5. Знайти максимальний та обчислити добуток від'ємних парних елементів у кожному з рядків матриці.

6. Знайти максимальний та обчислити кількість від'ємних додатних елементів у кожному із стовпчиків матриці.

7. Знайти максимальний та обчислити суму додатних парних елементів у кожному з рядків матриці .

8. Знайти максимальний та обчислити кількість від'ємних парних елементів у кожному із стовпчиків.

9. Знайти мінімальний та обчислити суму від'ємних парних елементів у кожному з рядків матриці.

10. Знайти мінімальний та обчислити добуток непарних елементів у кожному із стовпчиків матриці.

11. Знайти мінімальний та обчислити кількість нульових елементів у кожному з рядків матриці.

12. Знайти т мінімальний а обчислити добуток додатних елементів у кожному із стовпчиків матриці.

13. Знайти мінімальний та обчислити суму ненульових елементів у кожному з рядків матриці.

14. Знайти мінімальний та обчислити суму додатних непарних елементів у кожному із стовпчиків матриці.

15. Знайти мінімальний та обчислити кількість від'ємних елементів у кожному з рядків.

16. Знайти мінімальний та обчислити кількість ненульових елементів у кожному із стовпчиків матриці.

17. Знайти мінімальний та обчислити суму від'ємних елементів у кожному з рядків матриці.

18. Знайти мінімальний та обчислити добуток ненульових елементів у кожному із стовпчиків.

19. Знайти т мінімальний а обчислити кількість додатних парних елементів у кожному з рядків матриці.
20. Знайти мінімальний та обчислити середнє арифметичне непарних елементів у кожному із стовпчиків матриці.
21. Знайти максимальний та обчислити середнє арифметичне додатних елементів у кожному із стовпчиків матриці.
22. Знайти максимальний та обчислити кількість додатних непарних елементів у кожному з рядків . матриці
23. Знайти максимальний та обчислити кількість від'ємних парних елементів у кожному із стовпчиків.
24. Знайти т максимальний а обчислити середнє арифметичне від'ємних елементів у кожному з рядків матриці.
25. Знайти максимальний та обчислити кількість додатних непарних елементів у кожному із стовпчиків матриці
26. Знайти максимальний та обчислити середнє арифметичне від'ємних непарних елементів у кожному з рядків матриці.
27. Знайти максимальний та обчислити суму додатних непарних елементів у кожному із стовпчиків матриці.
28. Знайти максимальний та обчислити добуток додатних елементів у кожному з рядків матриці.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Дати характеристику динамічної області пам'яті.
- 2 Які операції та функції виділяють та звільняють пам'ять в динамічній області?
- 3 Яке призначення та правила використання операції new?
- 4 Яке призначення та правила використання операції delete?
- 5 Яке призначення та правила використання функції malloc()?

- 6 Чим відрізняється операція new від функції malloc()?
- 7 Яке призначення та правила використання функції free()?
- 8 Дати характеристику динамічного масиву даних.
- 9 Описати спосіб виділення пам'яті під динамічну матрицю.

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 13
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм з використанням функцій

Мета: Придбати практичні навички визначення функції, передачі параметрів в функцію «за значенням», виклику функції. Придбати практичні навички в організації меню користувача з використанням оператора switch.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Опис та виклик функції. Передача параметрів за значенням.

Зі збільшенням обсягу програми стає неможливим утримувати в пам'яті всі її деталі. Природним способом боротьби зі складністю будь-якого завдання є розбивка його на частини. У C++ завдання може бути розділена на більше прості й зрозумілі за допомогою функцій, після чого програму можна розглядати на рівні взаємодії функцій. Це важливо, оскільки людина здатна пам'ятати обмежене число фактів. Використання функцій є першим кроком до підвищення ступеня абстракції програми й веде до спрощення її структури.

Поділ програми на функції дозволяє також уникнути написання аналогічних частин коду, оскільки функцію записують один раз, а викликати її на виконання можна багаторазово з різних крапок програми.

Функцію можна розглядати як операцію, визначену користувачем. У загальному випадку вона задається своїм ім'ям. Функція може приймати параметри й повертати значення.

Будь-яка програма в C++ складається з функцій, одна з яких повинна мати ім'я `main` (з неї починається виконання програми). Функція починає виконуватися в момент виклику. Будь-яка функція повинна бути оголошена й визначена. Як і для інших величин, оголошень може бути декілька, а визначення тільки одне. Оголошення функції повинне перебувати в тексті раніше її виклику для того, щоб компілятор міг здійснити перевірку правильності виклику.

В оголошенні функції (прототипі, заголовку, сигнатурі) описується її інтерфейс. Він містить всі дані про те, яку інформацію повинна одержувати функція (список параметрів) і яку інформацію вона повертає. Визначення функції містить, крім оголошення, тіло функції, що складається з послідовності операторів й описів у фігурних дужках:

тип ім'я ([список_параметрів])


```
{тіло_функції}
```

тип – це тип значення, що повертається функцією. Він може бути будь-яким, крім масиву й функції (але може бути покажчиком на масив або функцію). Якщо функція не повинна повертати значення, вказується тип `void`.

список параметрів визначає величини, які потрібно передати у функцію при її виклику. Елементи списку параметрів розділяються комами. Для кожного параметра, переданого у функцію, вказується тип й ім'я.

тіло функції - дії, які робить функція.

У визначенні, в оголошенні й при виклику однієї й тієї ж функції порядок проходження параметрів повинні збігатися.

Тип значення, що повертається і типи параметрів спільно визначають тип функції.

Повернення значення з функції реалізується оператором

```
return [вираз];
```

Якщо функція описана як `void`, вираз не вказується.

Приклади:

```
int f1() {return 1;}    //правильно
void f2() {return 1;}  //неправильно, f2 не повинна повертати значення
double f3 {return 1;} //правильно, 1 перетвориться до типу double
```

Для виклику функції потрібно вказати її ім'я, за яким у круглих дужках через кому перераховуються імена переданих аргументів. Виклик функції може перебувати в будь-якому місці програми, де по синтаксисі припустиме вираження того типу, що формує функція. Якщо тип функції не `void`, вона може входити до складу виразів або, в окремому випадку, розташовуватися в правій частині оператора присвоювання.

Приклад функції, що повертає суму двох цілих величин:

```
#include <iostream.h>
int sum (int a, int b); //оголошення функції - прототип
int main()
{
    int a=2, b=3, c,d;
    c=sum(a,b);          //виклик функції
    cin>>d;
    cout<<sum(c,d);      //виклик функції
    return 0;
```

```

}
int sum(int a, int b)           //визначення функції
{
    return (a+b);
}

```

При спільній роботі функції повинні обмінюватися інформацією. Це можна здійснити за допомогою глобальних змінних, через параметри й через значення, що повертається функцією.

Глобальні змінні видні у всіх функціях, де не описані локальні змінні з тими ж іменами, тому використати їх для передачі даних між функціями дуже легко. Проте , це не рекомендується, оскільки утрудняє налагодження програми й перешкоджає приміщенню функцій у бібліотеки загального користування. Потрібно прагнути до того, щоб функції були максимально незалежні, а їхній інтерфейс повністю визначався прототипом функції.

Параметри, перераховані в заголовку опису функції, називаються *формальними*, а записані в операторі виклику функції - *фактичними*.

При виклику функції в першу чергу обчислюються вирази, що розміщуються на місці фактичних параметрів, потім у стеці виділяється пам'ять під формальні параметри функції відповідно до їх типу, і кожному з них привласнюється значення відповідного фактичного параметра. При цьому перевіряється відповідність типів й, при необхідності, виконується їхнього перетворення. При невідповідності типів видається діагностичне повідомлення.

2 Оператор перемикач (оператор switch)

Оператор if дозволяє здійснити вибір тільки між двома варіантами.

Для того, щоб проводити вибір один з декількох варіантів використовується оператор switch.

Загальна форма запису:

```

switch (константний_вираз)
{
    case константне_значення1: оператор1; [оператори;] [break;]
    case константне_значення2: оператор2; [оператори;] [break;]
    . . .
}

```

```
case константне_значенняN: операторN; [оператори;][break;]  
[default: оператор;]  
}
```

Оператор виконується таким чином:

- 1 Обчислюється вираз в дужках оператора switch;
- 2 Обчислене значення порівнюється з мітками (константними значеннями) в опціях case;
- 3 Порівняння проводиться до тих пір, поки не буде знайдена мітка, відповідна даному значенню, після цього виконується оператор відповідної гілки;
- 4 Якщо відповідна мітка не знайдена, то виконується оператор в опції default.

Альтернатива default може бути відсутньою, тоді не буде проведено ніяких дій.

Оператор break здійснює вихід з оператора switch і перехід до наступного за ним оператора. За відсутності оператора break виконуватимуться всі оператори, починаючи з поміченого даною міткою і кінчаючи оператором в опції default.

Константний вираз (вираз, операнди якого константи) повинен бути цілого типу (включаючи char).

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Функції $Y(X)$ та $Z(X)$ задані формулами, що наведені нижче. За бажанням користувача обчислювати:

1. Значення тільки функції $Y(X)$;
2. Значення тільки функції $Z(X)$;
3. Значення обох цих функцій

для введеного користувачем дійсного аргументу X . Програма повинна задовольняти наступним вимогам:

- Програма повинна містити функцію забезпечення коректного введення користувачем аргументу X ;

- програма повинна містити 3 функції: 1) - для обчислення значення тільки функції $Y(X)$, 2) - для обчислення значення тільки функції $Z(X)$, 3) – для обчислення значення обох цих функцій;
- організувати меню для вибору користувачем варіанту обчислення значень функцій. Забезпечити повернення в меню програми після виконання чергового пункту меню. Вихід із програми здійснювати по відповідному пункту меню: «Выход из программы»;
- при обчисленні значень функцій враховувати область визначення функції. При введенні користувачем значення аргументу, при якому функція не визначена – вивести відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Варіант	Формула для $Y(X)$	Формула для $Z(X)$	Варіант	Формула для $Y(X)$	Формула для $Z(X)$
1	$Y(X) = x - 1$	$Z(X) = \frac{1}{\sqrt{x} - 1}$	15	$Y(X) = \frac{1}{5x - \sqrt{x}}$	$Z(X) = 8 - x$
2	$Y(X) = \sqrt{x} + 1$	$Z(X) = \frac{1}{2 + \sqrt{x}}$	16	$Y(X) = \frac{8}{4x + 1}$	$Z(X) = \frac{x}{4}$
3	$Y(X) = (x - 1)^2$	$Z(X) = \frac{1}{2 + x^2}$	17	$Y(X) = \frac{3x}{1 - \sqrt{x}}$	$Z(X) = \frac{1}{2x}$
4	$Y(X) = \sqrt{-1 + x}$	$Z(X) = \frac{5x}{x - 1}$	18	$Y(X) = x - 5$	$Z(X) = \frac{1}{\sqrt{x} + 1}$
5	$Y(X) = \frac{2 - \sqrt{x}}{4}$	$Z(X) = 2 + x$	19	$Y(X) = \frac{3x}{2}$	$Z(X) = \frac{1}{3 + x}$
6	$Y(X) = 1 - x$	$Z(X) = \frac{x - 1}{\sqrt{x}}$	20	$Y(X) = \sqrt{2 - x}$	$Z(X) = \frac{(1 + x)}{2x}$

7	$Y(X) = \frac{(x-1)^2}{2}$	$Z(X) = \frac{2x}{\sqrt{x}-1}$	21	$Y(X) = \frac{(x-2)^2}{5}$	$Z(X) = \frac{3-x}{4+\sqrt{x}}$
8	$Y(X) = 3x$	$Z(X) = \frac{1}{5-x}$	22	$Z(X) = \frac{1}{7x}$	$Y(X) = \frac{1}{4+x^3}$
9	$Y(X) = 3-x$	$Z(X) = \frac{2}{5x}$	23	$Z(X) = \frac{2x}{x-2}$	$Y(X) = \frac{1}{\sqrt{-5+2x}}$
10	$Y(X) = x-2$	$Z(X) = \frac{\sqrt{x-1}}{3}$	24	$Z(X) = (x-2)^2$	$Z(X) = \frac{1-x}{4x}$
11	$Y(X) = x-2$	$Z(X) = \frac{\sqrt{x-1}}{3}$	25	$Y(X) = 1+3x$	$Z(X) = \frac{x}{(1-x)^2}$
12	$Y(X) = \frac{1}{2+x^2}$	$Z(X) = \frac{3x}{1-x}$	26	$Y(X) = 3 - \frac{x}{5-x}$	$Z(X) = \frac{2}{\sqrt{x-4}}$
13	$Y(X) = 8x$	$Z(X) = 10 - \sqrt{x}$	27	$Z(X) = \frac{2+x}{x-2}$	$Y(X) = \frac{1}{\sqrt{-5+2x}}$
14	$Y(X) = \sqrt{1-x}$	$Z(X) = (x-1)^3$	28	$Y(X) = \sqrt{x} + 1$	$Z(X) = \frac{\sqrt{x+4}}{2+\sqrt{x}}$

КОНТРОЛЬНІ ПИТАННЯ

- 1 Дайте визначення поняття "функція"?
- 2 Дайте визначення поняття "прототип (оголошення) функції"?
- 3 Дайте визначення поняття "виклик функції"?
- 4 Дайте визначення поняття "визначення функції"?
- 5 Дайте визначення поняття "формальний параметр"?
- 6 Дайте визначення поняття "фактичний параметр"?
- 7 Яким чином повертаються та передаються значення у функцію?
- 8 Яка роль оператора return при поверненні значень функцією?
- 9 Які можуть бути типи даних, що повертаються функцією?
- 10 Навіщо використовувати прототип функції до моменту її використання?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.

2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>

3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 14
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм з використанням функцій, які приймають параметри-масиви

Мета: Придбати практичні навички визначення та виклику функцій з передачею параметрів за посиланням, використання масивів в якості параметрів функцій.

Час 4 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

1 Передача масивів як параметрів функції

При використанні масиву як параметра функції, у функцію передається покажчик на його перший елемент, тобто масив завжди передається за адресою. При цьому губиться інформація про кількість елементів у масиві, тому розмірність масиву варто передавати як окремий параметр. Тому що у функцію передається покажчик на початок масиву (передача за адресою), то масив може бути змінений за рахунок операторів тіла функції.

Приклад:

В заданому лінійному масиві цілих чисел видалити всі парні елементи.

Програма, наведена в лістингу 1, містить три функції:

1 `int form(int *a)` – функція формування псевдодинамічного масиву і заповнення його введенням з клавіатури. Параметр `int *a` – це покажчик на перший елемент масиву. Оскільки масив передається в функцію за посиланням, функція може змінювати масив, а саме заповнювати його значеннями введенням з клавіатури. Функція повертає введену користувачем реальну кількість елементів в масиві.

2 void print(int *a, int n) – функція виведення масиву на екран. Параметр int *a – це покажчик на перший елемент масиву. Параметр int n – це кількість елементів в масиві (реальний розмір масиву)

3 void Dell(int* a, int *n) – функція видалення з масиву парних елементів. Оскільки розмір масиву при цьому міняється, то кількість елементів в масиві передається за посиланням (параметр int *n).

Лістинг 1.

```
#include <iostream.h>
#include <conio.h>
int form(int *a)
{
    int n;
    cout<<"\nВведите количество элементов в массиве ";
    cin>>n;
    for(int i=0;i<n;i++)
        cout<<"Введите элемент A["<<i+1<<" ] ";
        cin>>a[i];
        cout<<endl;
    return n;
}
void print(int *a,int n)
{
    for(int i=0;i<n;i++)
        cout<<a[i]<<" ";
    cout<<"\n";
}

void Dell(int* a, int *n)
{
    int j=0,i,b[100];
    for(i=0;i<n;i++)
        if(a[i]%2!=0)
        {
            b[j]=a[i];j++;
        }
    n=j;
    for(i=0;i<n;i++) a[i]=b[i];
}

void main()
```



```

{
    int a[100];
    int n;
    n=form(a);
    cout<<"Введённый массив:"<<endl;;
    print(a,n);
    Dell(a,&n);
    cout<<"Изменённый массив:"<<endl;;
    print(a,n);
}

```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Обробити згідно завдання Вашого варіанту псевдодинамічний масив даних, використовуючи функції, які приймають масив як якість параметру. Програма повинна задовольняти наступним вимогам:

- програма повинна містити наступні функції:
 - 1 введення значень елементів масиву;
 - 2 функції, що відповідають пунктам а) та б) завдання;
 - 3 виведення значень елементів масиву, яка буде викликатись з функції, що відповідає пункту б) завдання;
- організувати меню для вибору користувачем варіанту виконання вказаних вище функцій. Забезпечити повернення в меню програми після виконання чергового пункту меню. Вихід із програми здійснювати по відповідному пункту меню: «Выход из программы»;
- при обробці масиву враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

№	Постановка завдання
1	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) добуток елементів масиву, розташованих між максимальним та мінімальним елементами; б) впорядкувати елементи масиву по зростанню, використовуючи сортування обмінами.
2	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) добуток елементів масиву, розташованих між максимальним по модулю та мінімальним по модулю елементами; б) впорядкувати елементи масиву за зменшенням, використовуючи метод вибору.
3	У одномірному масиві, який складається з n цілих елементів, обчислити: а) суму елементів масиву, розташованих між першим та останнім нульовими елементами; б) перетворити масив таким чином, щоб спочатку розташовувались усі додатні елементи, а потім – усі від’ємні (елементи, які дорівнюють 0, рахувати додатними).
4	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих до останнього додатного елементу;

№	Постановка завдання
	б) стиснути масив, видалив з нього усі елементи, модуль яких знаходиться в інтервалі $[a,b]$. Елементи, що задовольняють вказаній умові замінити нулями та розмістити в кінці масиву.
5	У одномірному масиві, який складається з n цілих елементів, обчислити: а) добуток елементів масиву, розташованих між першим та другим нульовими елементами; б) перетворити масив таким чином, щоб у першій його половині розташовувались елементи, що стоять у непарних позиціях, а у другій половині – елементи, що стоять у парних позиціях.
6	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих після першого додатного елементу; б) стиснути масив, видаливши з нього усі елементи, модуль яких не знаходиться в інтервалі $[a,b]$. Елементи, що задовольняють вказаній умові замінити нулями та розмістити в кінці масиву.
7	У одномірному масиві, який складається з n цілих елементів, обчислити: а) добуток елементів масиву, розташованих між передостаннім та останнім нульовими елементами; б) перетворити масив таким чином, щоб у першій його половині розташовувались парні елементи, а у другій половині – непарні елементи.
8	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих між передостаннім та останнім додатними елементами; б) перетворити масив таким чином, щоб елементи, які дорівнюють нулю, розташовувались після всіх інших.
9	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих між першим та другим від'ємним елементами; б) перетворити масив таким чином, щоб елементи, які дорівнюють заданому користувачем числу, розташовувались після всіх інших.
10	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму модулів елементів масиву, розташованих після першого від'ємного елемента;

№	Постановка завдання
	б) стиснути масив, видалив з нього усі максимальні. Елементи, що задовольняють вказаній умові, замінити нулями та розмістити в кінці масиву.
11	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих після першого максимального елемента; б) впорядкувати елементи масиву за зменшенням модулів елементів, використовуючи сортування вставками.
12	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму модулів елементів масиву, розташованих після останнього максимального елемента; б) стиснути масив, видалив з нього усі мінімальні елементи. Елементи, що задовольняють вказаній умові, замінити нулями та розмістити в кінці масиву.
13	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) добуток елементів масиву, розташованих після першого мінімального елемента; б) впорядкувати елементи масиву за зменшенням модулів елементів, використовуючи сортування обмінами.
14	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) добуток елементів масиву, розташованих після першого максимального по модулю елемента; б) перетворити масив таким чином, щоб спочатку розташовувались усі від'ємні елементи, а потім – усі додатні (елементи, що дорівнюють 0, рахувати додатними).
15	У одномірному масиві, який складається з n цілих елементів, обчислити: а) суму елементів масиву, розташованих після останнього елемента, який дорівнює 0; б) перетворити масив таким чином, щоб спочатку розташовувались усі елементи, ціла частина яких не перевищує 1, а потім – усі інші.
16	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) добуток елементів масиву, розташованих до першого елемента з вказаним користувачем значенням;

№	Постановка завдання
	б) перетворити масив таким чином, щоб спочатку розташовувались усі елементи, що більше заданого користувачем числа, а потім – усі інші елементи.
17	У одномірному масиві, який складається з n цілих елементів, обчислити: а) суму елементів масиву, розташованих до першого елемента, який дорівнює 0; б) перетворити масив таким чином, щоб спочатку розташовувались усі додатні парні елементи, а потім – усі інші.
18	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму додатних елементів масиву, розташованих між першим та останнім нульовими елементами; б) змінити порядок проходження елементів у масиві на зворотній.
19	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих після першого елемента з заданим користувачем значенням; б) впорядкувати елементи масиву за зменшенням модулів елементів, використовуючи метод вибору.
20	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, значення яких не дорівнює введеному користувачем числу; б) впорядкувати елементи масиву за збільшенням модулів елементів, використовуючи сортування вставками.
21	У одномірному масиві, який складається з n цілих елементів, обчислити: а) суму елементів масиву, розташованих між першим та останнім нульовими елементами; б) перетворити масив таким чином, щоб спочатку розташовувались усі від'ємні непарні елементи, а потім – усі інші.
22	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих до останнього додатного елемента; б) стиснути масив, видалив з нього усі елементи, модуль яких знаходиться в інтервалі $[a, b]$. Елементи, що задовольняють вказані умови, замінити нулями та розмістити в кінці масиву.
23	У одномірному масиві, який складається з n цілих елементів, обчислити: а) суму елементів масиву, розташованих після останнього елемента, який дорівнює 0; б) перетворити масив таким чином, щоб спочатку розташовувались усі елементи, ціла частина яких не перевищує 1, а потім – усі інші.
24	У одномірному масиві, який складається з n дійсних елементів, обчислити:

№	Постановка завдання
	а) суму додатних елементів масиву, розташованих до максимального елемента; б) змінити порядок проходження елементів у масиві на зворотній.
25	У одномірному масиві, який складається з n цілих елементів, обчислити: а) суму елементів масиву, розташованих після максимального елемента; б) впорядкувати елементи масиву за збільшенням модулів елементів, використовуючи сортування обмінами.
26	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму модулів елементів масиву, розташованих після першого від'ємного елемента; б) стиснути масив, видалив з нього усі елементи, значення яких не належать інтервалу $[a, b]$. Елементи, що задовольняють вказані умови, замінити нулями та розмістити в кінці масиву.
27	У одномірному масиві, який складається з n цілих елементів, обчислити: а) добуток елементів масиву, розташованих між першим та другим нульовими елементами; б) перетворити масив таким чином, щоб у першій його половині розташовувались елементи, що стоять у непарних позиціях, а у другій половині – елементи, що стоять у парних позиціях.
28	У одномірному масиві, який складається з n дійсних елементів, обчислити: а) суму елементів масиву, розташованих між першим та другим додатними елементами; б) перетворити масив таким чином, щоб елементи, які дорівнюють нулю, розташовувались після всіх інших.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Чим відрізняється передача параметрів за значенням та за посиланням?
- 2 Коли виникає необхідність передачі параметрів по посиланню?
- 3 При виклику функції яку операцію слід застосовувати до фактичних параметрів, що передаються по посиланню? Коли без цієї операції можна обійтись?
- 4 В тілі функції яку операцію слід застосовувати до фактичних параметрів, що передаються по посиланню? Коли без цієї операції можна обійтись?

- 5 Яка особливість передачі параметру – масиву?
- 6 В чому відмінність передачі як параметра лінійного масиву від матриці?

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 15
з дисципліни Технології (Програмування)

Тема: Реалізація програм обробки рекурсивних математичних функцій.

Мета: Придбати практичні навички реалізації рекурсивних функцій, визначенні термінальної умови та рекурентного співвідношення при обчисленні рекурсивних математичних функцій.

Час 2 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Приставаючи до рішення завдань, варто згадати, що:

- для передачі даних у функцію треба використати тільки параметри. Глобальні змінні, тобто змінні, оголошені поза функцією, використати не рекомендується;
- тип кожного фактичного параметра (константи або змінної) в інструкції виклику функції повинен збігатися з типом відповідного формального параметра, зазначеного в оголошенні функції;
- якщо параметр функції використовується для повернення результату, то в оголошенні функції цей параметр повинен бути посиленням, а в інструкції виклику функції як фактичний параметр повинен бути зазначений адреса змінної.

Функція, що прямо або побічно викликає сама себе, називається рекурсивною.

Наприклад:

```
int rgcd( int v1, int v2 )
{
    if ( v2 != 0 )
        return rgcd( v2, v1%v2 ); // рекурентне співвідношення
    return v1; // термінальна умова
}
```


Така функція обов'язково повинна визначати умову закінчення (термінальну умову), у противному випадку рекурсія буде тривати нескінченно. Подібну помилку так іноді й називають - нескінченна рекурсія. Для `rgcd()` умовою закінчення є рівність нулю залишку.

Виклик `rgcd( 15, 123 )`; повертає значення 3.

Послідовність рекурсивних викликів цієї функції демонструє таблиця трасування виклику `rgcd(15,123)`

v1	v2	return
15	123	<code>rgcd(123,15)</code>
123	15	<code>rgcd(15,3)</code>
15	3	<code>rgcd(3,0)</code>
3	0	3

Останній виклик, `rgcd(3,0)`; задовольняє умові закінчення. Функція повертає найбільший загальний дільник, він же повертається й кожним попереднім викликом. Говорять, що значення спливає (*percolates*) нагору, поки керування не повернеться у функцію, що викликала `rgcd()` у перший раз.

Рекурсивні функції звичайно виконуються повільніше, ніж їх не рекурсивні (ітеративні) аналоги. Це пов'язане з витратами часу на виклик функції. Однак, як правило, вони компактніше й зрозуміліше.

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Розробити програму з використанням рекурсивних функцій згідно завдання Вашого варіанту. Програма повинна задовольняти наступним вимогам:

- враховувати область визначення функції. При введенні вхідних даних, що не входять в область визначення функції, повинно бути виведено

відповідне повідомлення (наприклад: На введенных входных данных последовательность не определена)

- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

1. Отримати m -й член послідовності: $\frac{x^{2m-1}}{(2m-1)!}$. Значення x та m вводяться користувачем.
2. Отримати k -й член послідовності: $\frac{(1-x)^{k+1}+1}{((k-1)!+1)^2}$. Значення x та k вводяться користувачем.
3. Отримати n -й член послідовності: $\frac{(-1)^n(n+1)}{n!}$. Значення n вводяться користувачем.
4. Отримати m -й член послідовності: $\frac{x^m}{(m+1)!}$. Значення x та m вводяться користувачем.
5. Отримати s -й член послідовності: $\frac{(-1)^s(\sin 3x)^s}{(s+2)!+(2s+1)!}$. Значення x та s вводяться користувачем.

6. Отримати k -й член послідовності: $\frac{(-1)^k (\cos 3x)^{k+1}}{(3k+1)!+3k!}$. Значення x та k вводяться користувачем.
7. Отримати s -й член послідовності: $\frac{(-1)^{2s-1} \cos(2x)^{s+1}}{(s+3)!-5}$. Значення x та s вводяться користувачем.
8. Отримати n -й член послідовності: $\frac{(-1)^n (\cos x)^{n+1}}{5+(n+2)!}$. Значення x та n вводяться користувачем.
9. Отримати k -й член послідовності: $\frac{(1-x)^{k+1}+1}{((k-1)!+(k+2)!)^2}$. Значення x та k вводяться користувачем.
10. Отримати k -й член послідовності: $\frac{(-2)^{k+1}+5}{4^k+5^{k+1}}$. Значення k вводяться користувачем.
11. Отримати z -й член послідовності: $\frac{(-1)^{2z+1}(\operatorname{tg} x)^{2z}}{(2z+1)!-(3z+2)!}$. Значення x та z вводяться користувачем.
12. Отримати n -й член послідовності: $\frac{(-1)^n (\lg x)^{n+1}}{(2n+2)!-15}$. Значення x та n вводяться користувачем.
13. Дано натуральне число n . Отримати добуток перших n співмножників: $\frac{1}{1} * \frac{3}{4} * \frac{5}{6} * \dots$. Значення n вводиться користувачем.
14. Дано натуральні числа n, k ($n \geq k \geq 0$). Обчислити $\frac{n(n-1)\dots(n-k+1)}{k!}$.
15. Дано дійсні числа x та y . Отримати найбільше ціле n , при якому $\frac{x^n}{2^n} < y$.
16. Дано дійсне число $\varepsilon < 1$. Нехай $y_1 = 1, y_2 = \frac{1+1}{1+2}, y_3 = \frac{\frac{2}{3}+1}{\frac{2}{3}+2}, \dots$. Знайти перший член y_n , для якого виконується нерівність $y_n - y_{n-1} < \varepsilon$.
17. Отримати n -й член послідовності: $\frac{(-1)^n x^n}{(n!+1)!}$. Значення x та n вводяться користувачем.

- 18.Отримати k-й член послідовності: $\frac{(-1)^k \cos 3 x^{k+1}}{(2k!+2)!}$. Значення x та k вводяться користувачем.
- 19.Отримати k-й член послідовності: 2, 4, 16, 256, ...Значення k вводяться користувачем.
- 20.Отримати n-й член послідовності: $\frac{(-1)^n \lg x^{n+1}}{(2n+2)!-15}$. Значення x та n вводяться користувачем.
- 21.Отримати m-й член послідовності: $\frac{x^{2m-1}}{(2m-1)!+5}$. Значення x та m вводяться користувачем.
- 22.Отримати k-й член послідовності: $\frac{(1-x)^{k+1}}{((k-1)!+1)^2}$. Значення x та k вводяться користувачем.
- 23.Отримати n-й член послідовності: $\frac{(-1)^n (n+1)^n}{n!}$. Значення n вводяться користувачем.
- 24.Отримати m-й член послідовності: $\frac{x^{m+1}}{(m+1)!}$. Значення x та m вводяться користувачем.
- 25.Отримати s-й член послідовності: $\frac{(-1)^s (\sin 3x)^s}{(s+2)!+(2s+1)!}$. Значення x та s вводяться користувачем.
- 26.Отримати k-й член послідовності: $\frac{(-1)^k (\cos 3 x)^k}{(3k+1)!+(3k)!}$. Значення x та k вводяться користувачем.
- 27.Отримати s-й член послідовності: $\frac{(-1)^{2s+1} \cos(2 x)^{s1}}{(s+3)!+5}$. Значення x та s вводяться користувачем.
- 28.Отримати n-й член послідовності: $\frac{(-1)^n (\cos x)^n}{5+(n+2)!}$. Значення x та n вводяться користувачем.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Яка функція називається рекурсивною?
- 2 Які умови є необхідними для правильної організації рекурсивних викликів функцій?
- 3 Що таке термінальна умова?
- 4 Навіщо використовується рекурентне співвідношення?
- 5 Сформулюйте термінальну умову та рекурентне співвідношення для обчислення n – го члена наступної послідовності: 2, 4, 16, 265, ...

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 16
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм обробки текстових файлів.

Мета: Придбати практичні навички обробки текстових файлів з використанням функцій `getc()` та `putc()`, `fgets()` та `fputs()`, `fprintf()` та `fscanf()`.

Час 4 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Функції послідовного текстового введення-виведення дозволяють:

- вводити символи або рядки символів із текстового потоку (файлу);
- виводити в потік (файл) символи або рядки символів.

1 Функції посимвольного читання-запису

Функція `getc()`.

Функція `getc()` зчитує символ з файлу. Вона повертає цілочисельну змінну, молодший байт якої містить уведений символ. Якщо виникла помилка, то старший байт цієї змінної дорівнює 0.

Якщо при читанні виявлений кінець файлу, функція `getc()` повертає константу EOF. Таким чином у C кінець файлу виявляється після зчитування.

Функція `putc()`.

Функція `putc()` записує символ у файл, відкритий за допомогою функції `fopen()`. Прототип цієї функції виглядає так:

```
int putc( int ch, FILE* fp)
```

Тут параметр `fp` являє собою покажчик файлу, а аргумент `ch` є символом, що підлягає виведенню. Покажчик файлу повідомляє функції `putc()` у який саме файл варто зробити запис. Незважаючи на те, що параметр `ch` має тип `int`, у файл записується лише молодший байт. Якщо функція `putc()` виконана успішно, вона

повертає символ, записаний нею у файл. У противному випадку вона повертає константу EOF.

2 Функції `fgets()` та `fputs()`

Для рядкового введення-виведення використовуються наступні функції:

1) `char* fgets(char* s, int n, FILE* f)`, де
`char* s` – адреса, за якою розміщуються зчитані байти (символи),
`int n` – кількість зчитаних байтів (символів),
`FILE* f` – покажчик на файл, з якого відбувається зчитування.

Зчитування байтів закінчується після отримання `n-1` байтів або при досягненні керуючого символу `'\n'` – ознаки кінця рядка. Керуючий символ теж передається в рядок `s`. Рядок в будь-якому випадку закінчується символом `'\0'`. При успішному завершенні зчитування функція повертає покажчик на зчитаний рядок. При помилці зчитування повертається `0 (NULL)`.

2) `int puts(char* s, FILE* f)`, де
`char* s` – адреса, за якою отримуються байти, які записуються в файл.
`FILE* f` – покажчик на файл, в який відбувається запис.

Символ кінця рядка (`'\0'`) в файл не записується. Функція повертає EOF, якщо при запису в файл виникла помилка. При успішному записі функція повертає додатне число.

3 Функції `fprintf()`, `fscanf()`.

Ці функції відрізняються від відповідних `printf()`, `scanf()` тільки тим, що додається один параметр - покажчик на FILE створений при виклику функції `fopen()`.

`fprintf()` – форматоване виведення файл, пов'язаний з потоком. При цьому файл повинен бути відкритий як текстовий, у режимі, що допускає запис.
Синтаксис:

```
int fprintf (FILE * Потік, Формат, Список_змінних);
```

`fscanf()` – форматоване читання значень змінних з файлу, пов'язаного з потоком. Файл повинен бути відкритий як текстовий у режимі, що допускає читання. Синтаксис:

```
int fscanf (FILE * Потік, const char* Формат, Список_адрес);
```

Приклад. Написати програму, що визначає, чи зустрічається в заданому текстовому файлі задана послідовність символів. Довжина рядка тексту не перевищує 80 символів, текст не містить переносів слів, послідовність не містить пробільних символів.

Вихідні дані й результати

1. Текстовий файл невідомого розміру, що складає з рядків довжиною не більше 80 символів. Оскільки переноси відсутні, можна обмежитися пошуком заданої послідовності в кожному рядку окремо. Отже, необхідно пам'ятати тільки один поточний рядок файлу. Для її зберігання виділимо строкову змінну довжиною 81 символ (додатковий символ потрібно для завершального нуля).

2. Послідовність символів для пошуку, що вводиться з клавіатури. Оскільки за умовою завдання вона не містить пробільних символів, її довжина теж не повинна перевищувати 80 символів, інакше пошук завершиться невдачею. Для її зберігання також виділимо строкову змінну довжиною 81 символ.

Результатом роботи програми є повідомлення або про наявність заданої послідовності, або про її відсутність.

Представимо варіанти повідомлень у програмі у вигляді строкових констант. Для зберігання довжини рядка будемо використати іменовану константу. Для роботи з файлом буде потрібно службова змінна відповідного типу.

Алгоритм рішення завдання:

1. Порядково зчитувати текст із файлу.
2. Для кожного рядка перевіряти, чи втримується в ній задана послідовність.
3. Якщо так, то надрукувати із про наявність заданої послідовності й завершити програму.
4. При нормальному виході із циклу надрукувати повідомлення про відсутність заданої послідовності й завершити програму.


```

#include <fstream.h>
#include <string.h>
int main ()
{
    const int len = 81;          //1
    char word[len], line[len];    //2
    cout << "Введите слово для поиска: ";
    cin >> word;
    ifstream fin("text.txt", ios::in | ios::nocreate); //3
    if (!fin) {cout << "Ошибка открытия файла." << endl; return 1};
//4
    while (fin.getline(line, len)) //5
    {
        if (strstr(line, word)) //6
        { cout <<"Присутствует!"<<endl; return 0;}
    } cout << " Отсутствует!" << endl; return 0; //7
}

```

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Введенням з клавіатури створити текстовий файл, що містить декілька рядків символів (кількість рядків визначає користувач). Обробити кожний рядок символів згідно завдання Вашого варіанту. Вивести оброблений текстовий файл на монітор.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

Зауваження: словом в рядку символів вважається підпоследовність символів, що не містить символу пробілу всередині себе.

1 Знайти у кожному з рядків кількість слів, що закінчуються заданим користувачем символом. Вивести той рядок (рядки), що містить найбільше шуканих слів.

2 Знайти скільки разів у кожному з рядків зустрічається заданий користувачем символ. Вивести рядок (рядки), в яких заданий користувачем символ зустрічається найчастіше.

3 Знайти та вивести кількість слів у кожному з рядків символів. Вивести той рядок (рядки), що містить найбільше слів.

4 Знайти у кожному з рядків кількість символів, відмінних від пробілу. Вивести рядок (рядки), в яких найбільше символів, відмінних від пробілу.

5 Знайти у кожному із рядків кількість слів, що містять в собі заданий користувачем символ. Вивести той рядок (рядки), що містить найбільше шуканих слів.

6 У кожному з рядків знайти довжину найдовшого слова. Вивести це слово (слова).

7 Для кожного рядка з'ясувати, чи є в ньому слова довжиною в один символ. Вивести той рядок (рядки), в якому найбільше таких слів.

8 Знайти у кожному з рядків кількість слів довжиною в N символів (N вводить користувач). Для рядків, які містять шукані слова, вивести останнє таке слово в рядку.

9 Знайти та вивести кількість слів у кожному з рядків символів. Вивести тільки ті рядки, що містять задану користувачем кількість слів.

10 Для кожного рядка в кожному слові видалити перший символ. Вивести оновлений масив рядків символів.

11 Знайти скільки разів у кожному з рядків зустрічається заданий користувачем символ. Вивести тільки ті рядки, в яких заданий символ зустрічається N разів (N вводить користувач).

12 У кожному з рядків підрахувати та вивести кількість слів, що закінчуються заданим користувачем символом. Вивести тільки той рядок, що містить найменшу (відмінну від 0) кількість таких слів.

13 У кожному з рядків з'ясувати, чи містить він символи арифметичних операцій. Вивести той рядок (рядки), що містить символи арифметичних операцій, але не містить символ '+'.

14 У кожному з рядків підрахувати скільки разів зустрічаються в ньому голосні букви латинського алфавіту. Вивести той рядок (рядки), в якому найбільше таких символів.

15 У кожному з рядків підрахувати кількість слів, що закінчуються першою буквою латинського алфавіту. Для рядків, які містять шукані слова, вивести перше таке слово в рядку.

16 Для кожного з рядків символів підрахувати, скільки в ньому слів довжиною в один символ. Вивести рядок, в якому найбільше таких слів.

17 Для кожного рядка підрахувати кількість слів, що містять в собі два, заданих користувачем символи. Вивести той рядок (рядки), в якому найбільше таких слів.

18 У кожному з рядків видалити зайві пробіли. Вивести оновлений масив рядків символів.

19 Знайти та вивести скільки слів у кожному з рядків символів. Вивести той рядок, що містить найменше (але не 0) слів.

20 У кожному з рядків замінити всі однакові символи, що йдуть поряд, замінити одним символом. Наприклад, рядок “ aghaahh ddd vvnvvv” замінити рядком “ aghah d vnv”. Вивести оновлений масив рядків символів.

21 Знайти у кожному з рядків кількість слів, що не містять заданий користувачем символ. Вивести той рядок (рядки), в якому найбільше таких слів.

22 Знайти скільки разів у кожному з рядків зустрічається слова, що починаються та закінчуються заданим користувачем символом.

23 У кожному з рядків підрахувати скільки разів зустрічаються в ньому голосні букви кирилиці. Вивести той рядок (рядки), в якому найменше (але не 0) таких символів.

24 Для кожного з рядків символів підрахувати, скільки в ньому слів довжиною в два символи. Вивести рядок, в якому найменше (але не 0) таких слів.

25 Для кожного рядка поміняти місцями перший та останній символ кожного слова. Вивести оновлений масив рядків символів.

26 Для кожного рядка після заданого користувачем символу вставити символ ‘/’. Вивести оновлений масив рядків символів.

27 Для кожного рядка в кожному слові останній символ переставити на початок слова. Вивести оновлений масив рядків символів.

28 Для кожного рядка видалити всі входження заданого користувачем символу. Вивести оновлений масив рядків символів.

КОНТРОЛЬНІ ПИТАННЯ

- 1 Пояснити роботу функції `getc()` посимвольного зчитування з текстового файлу.
- 2 Пояснити роботу функції `putc()` посимвольного запису в файл.
- 3 Пояснити роботу функції `fgets()` рядкового зчитування з текстового файлу.
- 4 Пояснити роботу функції `fputs()` рядкового запису в файл.
- 5 Пояснити особливості роботи функцій `fprintf()` та `fscanf()` при роботі з текстовими файлами.

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Методичні рекомендації
до виконання лабораторної роботи № 17
з дисципліни Технології (Програмування)

Тема: Складання та налагодження програм обробки файлів структур.

Мета: Придбати практичні навички при складання програм роботи з файлами структур з використанням функцій `fread()`, `fwrite()`, `fseek()` та `ftell()`.

Час 4 години

ТЕОРЕТИЧНІ ВІДОМОСТІ ДО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ

Функції роботи з бінарним потоком даних

Усякий потік перед звертанням до нього повинен бути відкритий за допомогою функції `fopen` для введення, виведення або виконання обох цих операцій у текстовому або двійковому режимі. Відкриваючи потік, функція `fopen` повертає покажчик типу `FILE`, що використовується для посилання на нього у всіх операціях введення/виведення або керування покажчиком поточної позиції.

Формат й опис аргументів функції :

```
FILE *fopen (pathname, type)
```

де `pathname` – ім'я файлу, що відкривається, `type` – тип доступу до файлу.

Параметр `type` являє собою рядок символів і визначає характер доступу до файлу.

При роботі з бінарними файлами слід використовувати наступні типи доступу:

"rb" - існуючий бінарний файл відкрити для читання;

"wb" - бінарний файл відкрити для запису (у випадку існування файлу його вміст руйнується);

"ab" - бінарний файл відкрити для запису в кінець файлу (у випадку відсутності файл створюється знову);

"r+b" - існуючий файл відкрити для читання й запису;

"w+b" - бінарний файл відкрити для читання й запису (у випадку існування файлу його вміст руйнується).

Функція повертає покажчик типу FILE при нормальному завершенні операції й NULL у випадку виникнення помилки.

Приклад використання:

```
#include <stdio.h>
void main()
{
    FILE *stream;
    if ((stream = fopen("data", "rb")) == NULL)
        printf("Помилка при відкритті файлу");
}
```

Після завершення роботи з потоком, його варто закрити, використовуючи функцію fclose. У протилежному випадку це зробить операційна система по закінченні роботи програми.

Формат й опис аргументів функції:

```
int fclose(stream)
FILE *stream;          /* Покажчик на відкритий файл */
```

Значення, що повертається, дорівнює нулю при нормальному завершенні операції й EOF у випадку виникнення помилки. Приклад використання:

```
#include <stdio.h>
void main()
{ FILE *stream;
  stream = fopen("data", "w+b");
  .....
  .....
  fclose(stream);
}
```

Операції читання й запису для потоку виконуються, починаючи з поточної позиції покажчика, зміщаючи останній на відповідне число позицій. Використовуючи функцію fseek, покажчик можна встановити в довільну позицію файлу перед виконанням чергової операції, а функція rewind переміщує покажчик на початок файлу. Прототипи функцій введення/виведення потоком, а також визначення відповідних констант, типів і структур поміщені у файл stdio.h, якому необхідно включити в програму за допомогою директиви препроцесора #include.

Функція `feof` перевіряє виникнення ситуації "кінець файлу" при введенні/виведенні потоком.

Формат й опис аргументів функції:

```
int feof(stream)
```

```
FILE *stream; /* Показчик на відкритий файл */
```

Значення, що повертається функцією, відмінно від нуля при спробі читання за межами файлу й дорівнює нулю в протилежному випадку. Приклад використання:

```
#include <stdio.h>
char string[100];
FILE *stream;
void process(char*);
main()
{ while (!feof(stream))
  if (fscanf(stream, "%s", string))
    process(string);
}
```

Функція `fseek` переміщає показчик поточної позиції файлу.

Формат й опис аргументів функції:

```
int fseek(stream, offset, origin)
```

```
FILE *stream; /* Показчик на відкритий файл */
```

```
long offset; /* Зсув у байтах від origin */
```

```
int origin; /* Початкова позиція показчика */
```

Ця функція переміщає показчик, пов'язаний з відкритим файлом, на `offset` байт уперед або назад відносно заданого параметром `origin` початку відліку. Останній може бути однією з наступних символічних констант, певних у файлі `stdio.h`:

`SEEK_SET` - початок файлу

`SEEK_CUR` - поточна позиція показчика

`SEEK_END` - кінець файлу

Помітимо, що припустимим є позиціювання покажчика за межами кінця файлу, однак його переміщення за початок файлу приводить до помилки. Значення, що повертається функцією, дорівнює нулю при нормальному завершенні операції й відмінно від нуля в протилежному випадку.

Приклад використання:

```
#include <stdio.h>
void main()
{ int    result;
  FILE   *stream;
  stream = fopen("data", "r");
  .....
  .....
  result = fseek(stream, 0L, SEEK_SET);
               .....
               .....
}
```

Функція `rewind` установлює покажчик поточної позиції на початок файлу.

Формат й опис аргументів функції:

```
void    rewind(stream)
FILE    *stream;      /* Покажчик на відкритий файл */
```

Ця функція не повертає ніякого значення в крапку виклику.

Приклад використання:

```
#include <stdio.h>
void main()
{ int    dat_1 = 1, dat_2 = 7, dat_3, dat_4;
  FILE   *stream;
  stream = fopen("data", "w+");
  /*      Запис даних у файл      */
  fprintf(stream, "%d %d", dat_1, dat_2);
  /*      Установка покажчика на початок      */
  rewind(stream);
  /*      Зчитування даних      */
  fscanf(stream, "%d %d", &dat_3, &dat_4);
               .....
               .....
}
```

Функція


```
size_t fread (void *buffer, size_t size, size_t count, FILE *stream);
```

зчитує count елементів довжиною size байтів в область, задану покажчиком buffer з потоку stream;

Функція

```
size_t fwrite (const void buffer, size_t size, size_t n, FILE stream);
```

записує n елементів довжиною size байтів з буфера, заданого покажчиком buffer у потік stream. Повертає число успішно записаних елементів звичайно воно = size, але при помилці воно може бути менше.

Тип size_t визначений у файлі stdio.h, означає «будь-який цілий». У байт, зайнятих цим «будь-яким цілим» можна одержати за допомогою функції sizeof(). Наприклад sizeof(int) - скільки байтів займає тип int.

Приклад:

```
char buf[256]; // масив символів
fwrite(buf, 256, 1, fp); // перед buf не ставиться &, тому що buf = &buf[0]
int k[10]; // масив цілих
fwrite(k, sizeof(int), 10, fp);
```

Функція

```
long int ftell(FILE *f);
```

визначає поточну позицію у файлі f і повертає, як довге ціле позитивне число байтів, відлічуване від початку файлу. Якщо виникла помилка, функція поверне значення -1.

Як правило, довільний доступ потрібен лише при роботі з бінарними файлами. Оскільки текстові файли можуть вміщати керуючі символи, кількість байтів, на яке слід виконати переміщення файлового курсору, може не відповідати очікуваному символу. Функція fseek () коректно працює з текстовим файлом лише тоді, коли позиція курсора була заздалегідь визначена функцією ftell() та макросом seek_set.

ЗМІСТ І ПОСЛІДОВНІСТЬ ВИКОНАННЯ ЗАВДАНЬ

1 Провести аналіз поставленої задачі.

Загальна постановка завдання:

Створити та реалізувати обробку файлу структур даних згідно Вашого варіанта.

Програма повинна задовольняти наступним вимогам:

- повинні бути реалізовані наступні функції обробки файлу структур:
 - 1 перегляд вмісту файлу;
 - 2 додавання нового запису в файл;
 - 3 реалізація запитів до файлу згідно з індивідуальними завданнями Вашого варіанту.
- організувати меню для вибору користувачем варіанту виконання вказаних вище функцій. Забезпечити повернення в меню програми після виконання чергового пункту меню. Вихід із програми здійснювати по відповідному пункту меню: «Выход из программы»;
- при обробці файла враховувати, що шукані елементи можуть бути відсутні. В цьому випадку вивести користувачеві відповідне повідомлення;
- введення та виведення вхідних та вихідних даних повинно містити необхідні для користувача повідомлення.

2 Побудувати блок-схему алгоритму рішення задачі згідно Вашого варіанту.

3 Розробити та налагодити програму рішення задачі мовою програмування C/C++.

4 Оформити звіт з лабораторної роботи.

Звіт повинен містити наступні розділи:

- 1 Постановка задачі.
- 2 Блок-схема алгоритму рішення задачі.
- 3 Текст програми мовою програмування C/C++.
- 4 Копія вікна виконання програми.
- 5 Висновок.

ВАРІАНТИ ЗАВДАНЬ ЛАБОРАТОРНОЇ РОБОТИ

№	Умова задачі	Індивідуальні завдання
1	Скласти програму, що містить відомості про вогнепальну зброю МВС області. Структура повинна складатися з полів: • модель; • табельний номер; • калібр; • маса; • ПІБ власника.	Видати наступну інформацію: 1. ПІБ власників, що мають найбільшу в області кількість одиниць зброї калібру X із вказанням їх характеристик; 2. Перелік зброї масою не менш за M калібру K. X, M та K – вводити з клавіатури.
2	Скласти програму, що містить відомості про асортимент продержаних авто у автопарку «Власник». Структура повинна складатися з полів: • Марку авто; • Колір; • Потужність; • Рік випуску; • Пробіг; • ПІБ власника.	Видати наступну інформацію: 3. Переліки найстаріших та найновіших авто марки M. 4. Марки авто, що мають потужність не менш за X останніх 5-х років випуску із вказанням їх власників. X, M – вводити з клавіатури.
3	Скласти програму, що містить відомості про соціально незахищені сім'ї міста. Структура повинна складатися з полів: • Категорія – малозабезпечений, чорнобилець, інвалід(група), багатодітний; • Кількість членів у сім'ї; • Район мешкання; • Вид помешкання – особистий дім, багатоповерхівка, барак, без помешкання.	Видати наступну інформацію: 3. Перелік сімей з найбільшою кількістю мешканців району X; 4. Перелік сімей з помешканням виду M категорії K. X, M , K – вводити з клавіатури
4	Скласти програму, що містить відомості про ДТП області. Структура повинна складатися з полів: • Район ДТП; • Дата; • ПІБ власника авто, що скоїв ДТП;	Видати наступну інформацію: 3. Кількість постраждалих у ДТП в районі X за у період часу від A до B(дата).

	Кількість постраждалих.	4. ПІБ винуватців ДТП, де постраждалих найбільша кількість та ДТП, де не було постраждалих. Х, М, К – вводити з клавіатури
5	Скласти програму, що містить відомості про проходження диспансеризації у коледжі. Структура повинна складатися з полів: - ПІБ студента; - Група; - ПІБ керівника групи; - Рік народження; - Стать; - Ознака – чи пройдено флюорографію; - Ознака – чи пройдено стоматолога; - Ознака – чи пройдено терапевта;	Видати наступну інформацію: Перелік груп, де відсоток студентів, що не пройшли чи пройшли не повністю диспансеризацію - найбільший; Перелік студентів статі К, що не пройшли вид огляду Х із вказанням групи. Х, К – вводити з клавіатури
6	Скласти програму, що містить відомості про асортимент канцтоварів у магазині «Олівець». Структура повинна складатися з полів: - назву виробу; - ціну; - країну- виробника; - кількість одиниць; - модель.	Видати наступну інформацію: 3) Ціни 2-х найдорожчих виробів країни-виробника Y і їх назви; 4) Перелік виробів цінової категорії від А до В у кількості не менш за Х одиниць. Х, А, В та Y – вводити з клавіатури.
7	Скласти програму, що містить відомості про здачу студентами 3 курсу відділення 1 сесії. Структура повинна складатися з полів: - Група; - ПІБ студента; - Відмітки по 2 іспитам; - Відмітки по 3 залікам; - Ознака участі в суспільній праці (активна \ неактивна); - База навчання (бюджет/ контракт)	Видати відомість нарахування студентам- бюджетникам групи Х стипендії. Студент, що отримав всі відмітки «5» і активно брав участь в суспільній праці, зараховується на підвищену стипендію - доплата 50%, неактивно брав участь – доплата 25%. Студенти, що отримали «4» і «5», зараховуються на звичайну стипендію. Студент, що отримав одну оцінку «3», але активно займався суспільною працею, також зараховується на

		звичайну стипендію, інакше зачислення не буде. X та розмір звичайної стипендії – вводити з клавіатури.
8	Скласти програму, що містить відомості про здачу студентами коледжу сесії. Структура повинна складатися з полів: ▪ індекс групи; ▪ ПІБ студента; ▪ вік; ▪ стать; ▪ Відмітки по п'яти залікам і іспитам.	Видати наступну інформацію: 3) ПІБ невестидаючих студентів з вказаним індексом групи статі Y та кількості заборгованостей; 4) Середній бал, отриманий кожним студентом групи X, і всієї групи в цілому. X, Y, Z – вводити з клавіатури.
9	Скласти програму, що містить відомості про наявність білетів і рейсах Аерофлоту м. Дн-ська. Структура повинна складатися з полів: ▪ номер рейса; ▪ пункт призначення; ▪ час вилита; ▪ час прибуття; ▪ признак – є топливна дозаправка чи ні; ▪ кількість вільних місць в салоні.	Видати наступну інформацію: 1) Час відправки літака в місто X; 2) Кількість вільних місць на рейс в місто X з часом відправки Y. 3) відомості про літаки, що без дозаправки знаходяться у повітрі більш за Z годин. X, Y – вводити з клавіатури.
10	Скласти програму, яка містить відомості про наявність взуття в магазині фірми «Підбор». Структура повинна складатися з полів: ▪ артикул; ▪ найменування; ▪ кількість; ▪ колір; ▪ виробник; ▪ матеріал; ▪ ціна однієї пари.	Видати наступну інформацію: 1) Асортиментний список взуття із матеріалу E з вказівкою наявності кількості пар кожної моделі. 2) Інформацію про взуття найменування Y кольору Z. Y, Z, E – вводити з клавіатури.
11	Скласти програму, яка містить відомості про п'ять дисциплін, які на вибір бажає слухати студент. Структура повинна складатися з полів: ▪ ПІБ студента; ▪ Індекс групи; ▪ 2 дисципліни; ▪ стать;	Надрукувати список студентів статі Y, бажаючих прослухати дисципліну X. Якщо кількість бажаючих перевищує 8 чоловік, то відібрати студентів, що мають більш високий середній бал успішності.

	■ середній бал успішності;	X, Y – вводити з клавіатури.
12	Скласти програму, яка містить відомості про відправку поїздів дальнього проходження з Дніпропетровського вокзалу. Структура повинна складатися з полів: ■ номер поїзда; ■ час відправки; ■ станція призначення; ■ ПІБ начальника составу; ■ Кількість вагонів; ■ наявність білетів.	Видати наступну інформацію: 1) Перелік. поїздів з номерами від А до В, де кількість вагонів мінімальна; 2) Наявність білетів на поїзд з номером Z. А, В, Z – вводити з клавіатури.
13	Скласти програму, яка містить відомості про співробітників коледжу. Структура повинна складатися з полів: ■ ПІБ робітника; ■ Назва відділення; ■ Рік народження; ■ Стаж роботи; ■ Посада; ■ Стать; ■ Оклад.	Видати наступну інформацію: 2) Список співробітників статі С пенсійного віку на сьогоднішній день. 2) Середній стаж працюючих в відділенні Х. 3) Список співробітників з найменшим окладом; Х, С – вводити з клавіатури.
14	Скласти програму, що зберігає дані про товари на складі. «Забудовник». Структура повинна складатися з полів: ■ назву товару; ■ кількість товару; ■ дату виготовлення; ■ дату прийому товару; ■ термін зберігання. ■ умови зберігання – температура, вологість.	Видати наступну інформацію: 3) Вивести дані про товар Х та перелік тих товарів, кількість яких перевищує N з терміном зберігання не менш за M. 4) Списки товарів (окремо), які повинні зберігатися при температурі від Н до К та при вологості не більш за 60 %. Х, N, M, H, K – ввести з клавіатури.
15	Скласти програму, що зберігає дані про препарати аптек у м. Дн-ськ. Структура повинна складатися з полів: ■ назва препарату; ■ наявність препарату ; ■ № аптеки; ■ Виробник; ■ ціна;	1) Вивести перелік препаратів за найвищою ціною в аптеці №N, що відпускаються без рецептів; 2) Вивести список препаратів виробника Х, ціна яких знаходиться у межах А-В.

	■ Ознака: відпускається препарат за рецептом чи ні.	X, N, A та B ввести з клавіатури.
16	Скласти програму, що зберігає дані про матеріальні цінності в аудиторіях коледжу. Структура повинна складатися з полів: • назва матеріальної цінності; • інвентарний номер; • номер аудиторії • ознака - полягає списанню чи ні. • ПІБ відповідального за матеріальні цінності цієї аудиторії	1. Вивести список усіх матеріальних цінностей, що полягають списанню в аудиторіях з номерами з номерами від N до M. 2. Вивести перелік усіх аудиторій, де є в наявності матеріальна цінність з назвою X з їх інвентарними номерами. 3. Вивести також список усіх матеріальні цінності відповідального A. N, M, X та A ввести з клавіатури.
17	Скласти програму, що веде облік товарів кіосків міста. Структура повинна складатися з полів • назва товару; • номери кіосків де товар в наявності (не більше 3); • ціна; • виробник товару. • гатунок товару. • Наявність ГМО.	1) Вивести перелік усіх кіосків, де в наявності товар X із вмістом ГМО. 2) Вивести перелік товарів гатунку Y за найвищою ціною у кількості не менш за N. X, Y, N та A ввести з клавіатури.
18	Скласти програму, що веде облік відвідувань занять студентів групи. Структура повинна складатися з полів: ■ ПІБ студента; ■ Кількість пропусків по дням тижня; ■ Відмітка, про причину пропусків (уважна чи неуважна) по дням тижня. ■ Група ■ Відділення.	1) Вивести перелік усіх студентів, що мають кількість неповажних пропусків занять у кількості від N до M годин, вище 40 годин. 2) Вивести також перелік груп відділення X, в яких найвища кількість прогулів. X, N та M ввести з клавіатури.
19	Скласти програму, що зберігає дані про товари на складі електротоварів «Мегамакс». Структура повинна складатися з полів: ■ назву товару;	1) Вивести перелік товарів, прийнятих у день A з потужністю не менш за B Ватт. 2) Вивести також дані про товари, що мають найменшу ти

	▪ кількість товару; ▪ дату прийому товару; ▪ потужність товару..	найбільшу кількість за цінами від N до M. M,N,A та B – ввести з клавіатури.
20	Скласти програму, що веде облік товарів продуктових магазинів. Структура повинна складатися з полів ▪ назва товару; ▪ номер магазину; ▪ ціна; ▪ термін зберігання. ▪ гатунок товару.	1) Вести дані про товари з найнижчим терміном зберігання 2)Вивести перелік магазинів, де є в наявності товар з назвою A гатунку B. X, A та B – ввести з клавіатури.
21	Скласти програму, що зберігає дані про продаж охолоджуючих напоїв у літні місяці по районах міста. Структура повинна складатися з полів: ▪ Район; ▪ Місяць; ▪ Загальну суму з продажу цього місяця. ▪ Загальну суму неліквіду цього місяця.	1) Обчислити відсоткове відношення продаж району X відносно загальної суми по всіх районах взагалі. 3)Вивести перелік районів з найменшою загальною сумою неліквіду району A за усі місяці. X, A – ввести з клавіатури.
22	Скласти програму обліку заборгованостей за електроенергію по районах міста за роки з 2008 по 2010. Структура повинна складатися з полів: ▪ Район міста; ▪ Суму заборгованості по трьом останнім рокам окремо (2008,2009, 2010).	1) Вивести перелік районів, у яких заборгованості з кожним роком зменшувалися. 2) Обчислити також рік, де заборгованість по району X була найбільшою. X – ввести з клавіатури.
23	Скласти програму, що містить дані про лікарів та пацієнтів міської лікарні «Швидкої допомоги». Структура повинна складатися з полів: • ПІБ пацієнта; • вік; • відділення (хірургія, гастро, шкіряне, терапевтичне); • ПІБ лікаря, • Діагноз хворого.	Видати наступну інформацію: 3. Діагнози пацієнтів віком старіше за 50 років лікаря X. 4. Перелік усіх пацієнтів відділення Y з діагнозом Z. X,Y,Z вводити з клавіатури.

24	Скласти програму, що містить дані про обслуговування периферійних пристроїв фірмою "Іонік". Структура повинна складатися з полів: • назва підприємства, що обслуговується; • модель пристрою, який обслуговується; • прізвище та ініціали майстра; • назву виявленої неполадки; • вартість усунення неполадки; • дата виконання.	Видати наступну інформацію: 1. Прізвище та ініціали майстра, який обслуговував найбільшу кількість пристроїв моделі X. 2. Кількість усіх виявлених неполадок за вказаним проміжком часу від Y до Z. X,Y,Z вводити з клавіатури.
25	Скласти програму, що містить дані про банківські вклади. Структура повинна складатися з полів: • ПІБ власника вкладу; • сума вкладу; • річний % нарахувань; • сума нарахованих процентів; • дата створення вкладу; • дата повернення вкладу.	Видати наступні інформацію: 1. Видати інформацію про загальну суму всіх нарахованих процентів закладами особи X. 2. Знайти та вивести на екран найбільший річний % нарахувань.
26	Скласти програму, містить інформацію про лікарів муніципального медичного закладу. Відомості про кожного лікаря включають: ☐ прізвище та ініціали лікаря; ☐ спеціалізація; ☐ № кабінету; ☐ графік роботи (день тижня, час початку роботи, час кінця роботи);	Програма повинна забезпечувати: ☐ видачу відомостей про лікарів заданої спеціалізації. Список повинно бути відсортовано за днями тижня; ☐ виведення графіка роботи введеного лікаря.
27	Відомості про кожну іграшку включають: – назва іграшки; – виробник; – вартість (xx грн. xx коп.); вікові границі (наприклад, від 2 до 5 років);	Програма повинна забезпечувати: – виведення відсортованого в алфавітному порядку списку іграшок, вартість яких не перевищує введеного значення і які підходять дітям вказаного віку; – виведення списків найдорожчих та найдешевших іграшок вказаного виробника;

28	Відомості про кожний товар включають: – назва товару; – виробник; – країна, що імпортує товар; – об'єм партії товару, що постачається; – дата постачання товару.	Програма повинна забезпечувати: – пошук країн, в які експортується введений користувачем товар та загальний об'єм його експорту; – пошук товарів, які мають найбільший попит в заданий період часу (вивести перші п'ять товарів, відсортувавши за назвою);
29	Відомості про кожного клієнта включають: – Ід клієнта; – прізвище та ініціали клієнта; – назва обладнання, виданого на прокат; – дата видачі; – дата повернення; – ознака повернення; – ознака оплати за прокат.	Програма повинна забезпечувати: – формування списку обладнання, яке знаходиться «на руках» у певного клієнта; – формування відсортованого в алфавітному порядку списку боржників.
30	Відомості про кожного клієнта включають: – П.І.Б. менеджера; – П.І.Б. клієнта; – марка та модель придбаного автомобіля; – вартість покупки; – % винагороди менеджеру за продаж.	Програма повинна забезпечувати: – формування відомостей про всіх клієнтів певного менеджера; – формування відомостей про найбільшу суму покупки та % винагороди певного менеджера;

КОНТРОЛЬНІ ПИТАННЯ

- 1 Сформулювати порядок роботи з файлами в С програмі.
- 2 Що таке файловий покажчик? Як описується файловий покажчик?
- 3 Сформулювати призначення та описати прототип функції `foren()`.

- 4 Перерахувати та пояснити відмінність між основними типами доступу до файлів.
- 5 Сформулювати призначення та описати прототип функції `fclose()`.
- 6 Пояснити роботу функції `fread()` блокового зчитування з бінарного файлу.
- 7 Пояснити роботу функції `fwrite()` блокового запису в файл.
- 8 Пояснити роботу функції `fseek()` довільного доступу до даних бінарного файлу.
- 9 Пояснити роботу функції `ftell()`.

ЛІТЕРАТУРА

- 1 О.Г. Трофименко. С++. Основи програмування. Теорія та практика: [Текст] / О.Г. Трофименко. – Одеса: "Фенікс", 2010. – 544 с.: іл.
- 2 Браян В. Керніган, Деніс М. Річі. . Мова програмування С. (друге видання) [Електронний ресурс] /<http://m.programming.in.ua/programming/c-language/227-book-programming-c-kernighan.html>
- 3 Методичні рекомендації для виконання лабораторних робіт з дисципліни «Програмування». – 2014. – Електронний ресурс.

Критерії оцінювання захисту лабораторної роботи

з дисципліни Технології (Програмування)

Захист лабораторної роботи вміщує:

- реалізацію програми мовою програмування C/C++;
- оформлення звіту з лабораторної роботи, згідно з вимогами, наданими в методичних рекомендаціях до виконання лабораторних робіт;
- відповідь студента на контрольні запитання викладача.

Максимальна кількість балів за виконання та захист лабораторної роботи – 6 балів.

Критерії оцінювання та система нарахування балів за виконання та захист лабораторної роботи наведено у таблиці № 1:

Таблиця № 1

<i>Види завдань</i>	<i>Кількість балів</i>	<i>Критерії оцінювання виконання завдання</i>	<i>Максимальна кількість балів</i>
Реалізація програми	2	Студент правильно в повному об'ємі реалізує програму згідно завдання лабораторної роботи	2
	1	Студент в повному об'ємі реалізує програму, але допускає логічні помилки. Або Студент правильно реалізує програму в об'ємі 50%-60%	
	0	Студент неправильно реалізує програму..	
Оформлення звіту	2	Студент правильно оформляє звіт з лабораторної роботи згідно до вимог, наданих в методичних рекомендаціях.	2
	1	Студент робить помилки при розробці блок-схеми алгоритму рішення задачі.	
	0	Студент неправильно або в неповному об'ємі оформляє звіт з лабораторної роботи згідно до вимог, наданих в методичних рекомендаціях.	
Відповідь на контрольне питання	2	Студент правильно в повному об'ємі відповідає на контрольний запит до лабораторної роботи.	2
	1	Студент допускає помилки при відповіді контрольний запит до лабораторної роботи.	
	0	Студент неправильно або зовсім не відповідає на контрольний запит до лабораторної роботи.	
Разом			6

Відповідність кількості набраних студентом балів оцінці за 4 – бальною шкалою оцінювання наведена в таблиці № 2.

Таблиця № 2

Бальна	Національна
6	відмінно
4-5	добре
3-4	задовільно
0 – 2	незадовільно