

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ЖИТОМИРСЬКИЙ ДЕРЖАВНИЙ ТЕХНОЛОГІЧНИЙ УНІВЕРСИТЕТ

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ ТА ІНДИВІДУАЛЬНІ ЗАВДАННЯ
ДЛЯ САМОСТІЙНОЇ РОБОТИ СТУДЕНТІВ
ЗАОЧНОЇ ФОРМИ НАВЧАННЯ З ДИСЦИПЛІНИ**

“ОСНОВИ ПРОГРАМУВАННЯ”

ЧАСТИНА 1

Житомир
2019

УДК 004.424

Марчук Г.В., Левківський В.Л., Чижмотря О.В. Методичні рекомендації та індивідуальні завдання для самостійної роботи студентів заочної форми навчання з дисципліни "Основи програмування". Ч.1 / Г. В. Марчук, В. Л. Левківський, О. В. Чижмотря. – Житомир: ЖДТУ, 2019. – 48 с.

Методичні рекомендації містять теоретичний матеріал, приклади та вказівки для виконання завдань самостійної роботи, пов'язаних з вивченням основ програмування мовою С. Методичні рекомендації призначені для студентів, які навчаються заочною формою за спеціальностями 121 "Інженерія програмного забезпечення" та 125 "Кібербезпека".

Рецензенти:

Сугоняк І.І. – к.т.н., завідувач кафедри комп'ютерних наук Житомирського державного технологічного університету

Сфремов Ю.М. – к.т.н., доцент кафедри інженерії програмного забезпечення Житомирського державного технологічного університету

Затверджено на засіданні кафедри інженерії програмного забезпечення.
Протокол № 4 від 25 лютого 2019 р.

УДК 004.424

Зміст

Вступ	4
Тема № 1.	5
<i>Введення та виведення інформації в мові програмування C. Правила запису арифметичних виразів.</i>	5
Тема № 2.	8
<i>Типи даних. Арифметичні вирази. Програмування лінійних алгоритмів.</i>	8
Тема № 3.	11
<i>Арифметичні операції. Декремент. Інкремент.</i>	11
Тема № 4.	15
<i>Програмування розгалужених алгоритмів. Умовний оператор if-else.</i>	15
Тема № 5.	18
<i>Програмування розгалужених алгоритмів Оператор вибору switch.</i>	18
Тема № 6.	21
<i>Програмування розгалужених алгоритмів. Оператори циклу for і while.</i>	21
Тема № 7.	23
<i>Програмування розгалужених алгоритмів Оператори циклу for, while, do-while.</i>	23
Тема № 8.	26
<i>Логічні операції та вирази.</i>	26
Тема № 9.	28
<i>Вкладені цикли.</i>	28
Тема № 10.	29
<i>Побітові операції.</i>	29
Тема № 11.	32
<i>Одномірні масиви.</i>	32
Тема № 12.	37
<i>Дослідження роботи одномірних масивів.</i>	37
Тема № 13.	40
<i>Сортування одновимірного масиву.</i>	40
Тема № 14.	44
<i>Опрацювання двовимірних масивів.</i>	44
Тема № 15.	47
<i>Робота з двовимірними масивами.</i>	47

Вступ

Методичні рекомендації призначені для виконання завдань самостійної роботи студентів заочної форми навчання з навчальної дисципліни "Основи програмування".

Методичні рекомендації містять опис робіт за 15 темами, кожна з яких складається з 15 варіантів завдань. Перед виконанням кожної роботи необхідно вивчити відповідний теоретичний матеріал. Приклади програм, що наведені у методичних рекомендаціях, слід розглядати лише як один із можливих варіантів розв'язання поставленої задачі.

Під час виконання робіт студент повинен продемонструвати:

- творчий підхід до розробки програмного засобу;
- навички програмування на мові високого рівня С.

Студент повинен вміти провести аналіз поставленої задачі, написати код програми та провести аналіз отриманого результату.

Варіант завдання самостійної роботи обирається відповідно до номера студента за списком групи.

Типовий порядок виконання робіт й методичні рекомендації до її виконання:

- 1) ознайомитися з методичними рекомендаціями до роботи;
- 2) провести аналіз поставленої задачі;
- 3) побудувати блок-схему
- 4) написати код програми (програма має починатися з коментарів, що пояснює її призначення та авторські права);
- 5) провести налагодження написаної програми;
- 6) відповісти на контрольні питання;
- 7) оформити звіт і здати викладачу.

Звіт з будь-якої роботи повинен містити:

1. Титульний лист відповідно зразка.
2. Опис виконаних завдань:
 - Умова завдання (завдань).
 - Графічне представлення алгоритму.
 - Лістинг програми.
 - План тестування та результати роботи програми на тестових даних.
3. Висновки по роботі з урахуванням усіх виконаних завдань:
 - аналіз отриманих результатів за кожним пунктом завдання;
 - аналіз результатів тестування програм.

Вимоги до оформлення звіту

Поля сторінки: ліве – 2,5 см, праве, верхнє й нижнє – 1,5 см; шрифт Times New Roman (14 пт), міжрядковий інтервал – множник 1,5.

Номери сторінок мають знаходитися у правому нижньому куті. Титульний лист не нумерується.

Тема № 1.

Введення та виведення інформації в мові програмування C. Правила запису арифметичних виразів.

Мета: ознайомитися з основними поняттями і структурою мови програмування C, навчитися користуватися функціями вводу та виводу та записувати арифметичні вирази мовою програмування.

Література:

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

Зміст роботи

Завдання 1. Написати програму, яка виводить на екран власне прізвище та ім'я.

Завдання 2. Написати програму, яка виводить на екран японські вірші.

Завдання 3. Записати вираз у лінійному вигляді за правилами алгоритмічної мови:

$a^2 + b^3$	$x_1x_2 - x_2x_3 - x_3x_2$	$b^2 - 4ac$	$1 - x^2$
$c \frac{ab}{b^2}$	$\frac{ab^2}{2} + abc$	$ab + \frac{ab^2}{4}$	$\frac{1}{a} + \frac{1}{d}$
$\frac{1}{ab^2 + (a - d)}$	$1 + \frac{a - b^2}{4a^{-3}}$	$\frac{1 + a^4 + 2d}{2a + 4ad - d^3}$	$\frac{(x + 1) - (x - 1)}{2x}$

Завдання 4. Напишіть програму для підрахунку виразу за формулою. Всі змінні приймають дійсне значення.

Варіант	Завдання 2	Завдання 4
1	"Прозрачный водопад"... Упала в светлую волну Сосновая игла.	$\frac{a}{b} - \frac{a}{c} + a^2$
2	Ясная луна. У пруда всю ночь напролет брожу, любуюсь...	$\frac{a}{c} - \frac{d}{b} - \frac{a - c}{cd}$
3	Яркий лунный свет! На циновку тень свою Бросила сосна.	$\frac{a}{c} + \frac{d}{b} + \frac{1}{e}$
4	Я шел по мосткам, и вдруг – Там, в глубине потока, Сквозят водяные цветы.	$\frac{ac^2}{b} + \frac{d}{b} + \frac{b^2}{a}$
5	Я прилег в тени, За меня толчет мой рис Горный ручеек.	$\frac{ac^2}{bd} - \frac{d - a^2}{b}$

6	Я поднялся на холм, Полон грусти - и что же: Там шиповник в цвету!	$\frac{adc^2 + a - d}{bd} - \frac{d}{a}$
7	Я наказал ребенка, Но привязал его к дереву там, Где дует прохладный ветер.	$\frac{a}{cd^2} + \frac{d}{b-a} + \frac{1}{e}$
8	Я и забыла, Что покрашены губы мои... Чистый источник!	$\frac{ad}{cd^2} - \frac{cd}{ba}$
9	Я встретил гонца на пути. Весенний ветер, играя, Раскрытым письмом шелестит	$\frac{a(1-d)}{cd^2} - \frac{d}{b+a}$
10	Я в полночь посмотрел: Переменила русло Небесная река.	$\frac{a(c-d)}{d^2} - \frac{a}{b+a}$
11	Это мой собственный снег! Каким он кажется легким На плетеной шляпе моей!	$\frac{a + (c-d)}{d^2} - \frac{a-5}{c^2}$
12	Эй, сова, Гляди веселее. Льет Весенний дождь.	$\frac{a - d(c/a - d)}{a + d} - \frac{5}{c^2}$
13	Чуть забрезжил рассвет. Ускользнув от бакланов, играют Рыбы в мелкой воде.	$\frac{(c/a - d)}{a/4 + d} - \frac{5d - b}{c - a^2}$
14	Чужих меж нами нет! Мы все друг другу братья Под вишнями в цвету.	$\frac{(c/a - d/a)}{d} - \frac{5d}{c}$
15	Чтоб соловья Достойно принять, с ограды Сметаем пыль.	$\frac{(c/b - d/a)}{dc^2} - \frac{d+5}{c}$

Методичні рекомендації

Функція **printf()** призначена для виведення інформації за заданим форматом. Синтаксис:

printf("рядок виводу"[, аргумент1[, аргумент2, [...]]]);

Першим параметром функції є, як правило текст і формати виводу значень змінних, імена яких вказані в параметрах, що залишилися.

Приклад: *printf("мама мила раму");*

Формати виводу значень змінних задаються за допомогою **специфікатора перетворення** - послідовність символів, що починаються зі знаку %.

При виводі числових значень найчастіше використовуються наступні специфікатори:

%i, %d - для виводу цілих чисел;

%f - для виводу чисел з плаваючою комою;

Приклад: `printf("a=%db==%dc==%d ",a,b,c);`

Для введення інформації використовується функція `scanf()`. Синтаксис :

`scanf("Рядок формату",&аргумент1[,&аргумент2[, ...]]);`

Приклад: `scanf("%d",&a);`

Оголошення змінної має наступний синтаксис:

[<специфікація класу пам'яті>]<тип><ім'я>[=<ініціалізатор>][,<ім'я>[=<ініціалізатор>] ...];

Приклад: `int a,b;` або `int a,b=4,c;`

Приклади запису арифметичних виразів мовою програмування:

$S = \frac{xy}{z}$	<code>S=x*y/z</code>
$S = \frac{x}{yz}$	<code>S=x/(y*z)</code> або <code>S=x/y/z</code>
$S = \frac{x+y}{z-5}$	<code>S=(x+y)/(z-5)</code>
$S = \frac{2,5x + (x+1) + 2z}{y + x^2z}$	<code>S=(2,5*x+(x+1)+2*z)/(y+x*x*z)</code>

Приклад виконання програми

Приклад 1	Приклад 2	Приклад 3
<pre>#include<stdio.h> #include<locale.h> void main() { setlocale(LC_ALL, "Rus"); printf("мама мыла раму"); }</pre>	<pre>#include<stdio.h> void main() { int a=1,b=6,c=7; printf("a==%d\nb==%d\n c==%d\n",a,b,c); }</pre>	<pre>#include<stdio.h> void main() { int a,b; printf("A="); scanf("%d",&a); printf("B="); scanf("%d",&b); printf("A+B=%d",a+b); }</pre>

Контрольні питання:

1. Що таке мова програмування? Описати види класифікації мов програмування.
2. Що таке тип даних?
3. Описати класифікацію типів даних в мові програмування C.
4. Описати структуру C програми.
5. Описати синтаксис функції `printf()`.
6. Описати синтаксис функції `scanf()`.

Тема № 2.

Типи даних. Арифметичні вирази. Програмування лінійних алгоритмів.

Мета: ознайомитися з поняттям і структурою мови програмування C, з величинами простих типів, їх властивостями та операціями над ними.

Література:

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

Зміст роботи

Завдання. Написати програму для обрахунку значення виразу при заданих вхідних даних. Отриманий результат порівняти з правильною відповіддю.

Варіант	Завдання
1	$s = \frac{2 \cos(x^2) - \frac{1}{2}}{\frac{1}{2} + \sin(y^{2-z})} + \frac{z^2}{7 - \frac{z}{3}}$ <p>При $x = 14,26$; $y = 5,7 \times 10^{-3}$; $z = 3,5 \times 10^{-2}$ відповідь $s = -3,621633836$</p>
2	$s = \frac{\sqrt[3]{3 + (x - y)}}{x^2 + z^3 + 4} - \operatorname{tg}(z)$ <p>При $x = -1,5$; $y = 1,75 \times 10^{-3}$; $z = 0,805 \times 10^2$ відповідь $s = -2,437012134$</p>
3	$s = \frac{1 + \sin(x + y)}{\left z - \frac{2x}{1 + x^2 y^2} \right } x^{ y } + \operatorname{tg}\left(\frac{1}{z}\right)$ <p>При $x = 2,74 \times 10^{-2}$; $y = -0,725$; $z = 0,13 \times 10^2$ відповідь $s = 0,079110706$</p>
4	$s = (\cos(x) - \sin(y))^{(1 + 2 \sin z)} \left(z + \frac{z^2}{2} + \frac{z^3}{3} + \frac{z^4}{4} \right)$ <p>При $x = 1,4 \times 10^2$; $y = -0,725$; $z = 13 \times 10^{-2}$ відповідь $s = 0,053139728$</p>
5	$s = \sin(y^{-\sqrt{ x }}) \left(z - \frac{x}{y} \right) + \sin(\cos(z))$ <p>При $x = 0,0004 \times 10^2$; $y = -7,7$; $z = 0,413 \times 10^2$ відповідь $s = -26,26284828$</p>
6	$s = \sqrt{10(\sqrt[3]{x} + x^{y+2})} \sin(z - x + y)$ <p>При $x = 0,00165 \times 10^2$; $y = 77,34$; $z = 1,3 \times 10^{-2}$ відповідь $s = -2,739595$</p>
7	$s = 5 \cos(x) - \frac{z}{4} \sin(y) \frac{4 z - x - 5}{z(x - y) + z^2}$ <p>При $x = 0,0134 \times 10^2$; $y = -7,33$; $z = 5,13 \times 10^{-4}$ відповідь $s = 1,556037$</p>

8	$s = \frac{e^{z+y}(z-y)^{x+z}}{\sin(x) + \sin(y)} + \sqrt[4]{x^7 + \ln(y)}$ <p>При $x=0,165$; $y=7,33$; $z=5,13 \times 10^{-4}$ відповідь $s=$</p>
9	$s = \left x^{\frac{y}{x}} - \sqrt[3]{\frac{y}{x}} \right + (y-x) \frac{\cos y - \frac{z}{(y-x)}}{1 + (y-x)^2}$ <p>При $x=1,765 \times 10^2$; $y=17,333$; $z=2,193 \times 10^{-2}$ відповідь $s=-32421,5703$</p>
10	$s = 2^{-x} \sqrt{x + \sqrt[4]{ y }} \sqrt[3]{e^{x-1/\sin z}}$ <p>При $x=3,765 \times 10^2$; $y=1,613 \times 10^3$; $z=0,613$ відповідь $s=$</p>
11	$s = y^{\sqrt[3]{ x }} + \cos^3(y) \frac{ x-y \left(1 + \frac{\sin^2 z}{\sqrt{x+y}} \right)}{e^{ x-y } + \frac{x}{2}}$ <p>При $x=7,765$; $y=0,714$; $z=19,103$ відповідь $s=0.007127$</p>
12	$s = 2^{(y^x)} + (3^x)^y - \frac{y \left(\operatorname{arctg} z - \frac{1}{3} \right)}{ x + \frac{1}{y^2 + 1}}$ <p>При $x=3,265$; $y=0,333$; $z=0,447 \times 10^{-4}$ відповідь $s=$</p>
13	$s = \frac{\sqrt[4]{y + \sqrt[3]{x-1}}}{ x-y (\sin^2 z + \operatorname{tg} z)}$ <p>При $x=16,165$; $y=7,433 \times 10^{-3}$; $z=8,13 \times 10^4$ відповідь $s=-0.032629$</p>
14	$s = \frac{y^{x+1}}{\sqrt[3]{ y-2 } + 3} + \frac{x + \frac{y}{2}}{2 x+y } (x+1)^{-1/\sin z}$ <p>При $x=1616,5 \times 10^{-3}$; $y=17,33$; $z=5,13 \times 10^2$ відповідь $s=$</p>
15	$s = \frac{x^{y+1} e^{y-1}}{1 + x y - \operatorname{tg}(z) } (1 + y-x) + \frac{ y-x ^2}{2} - \frac{ y-x ^3}{3}$ <p>При $x=2,444$; $y=0,869 \times 10^{-2}$; $z=-0,13 \times 10^3$ відповідь $s=-0,498707$</p>

Методичні рекомендації

Для використання математичних функцій необхідно включити заголовочний файл **math.h** - #include<math.h>.

Математична функція	Функція бібліотеки math.h	Опис
$ x $	abs(x) fabs(x)	Обрахування абсолютного значення
$\cos(x)$	cos(x)	Обрахування косинуса x
$\sin(x)$	sin(x)	Обрахування синуса x
$\text{tg}(x)$	tan(x)	Обрахування тангенса x
e^x	exp(x)	Обрахування експоненти числа x
\sqrt{x}	sqrt(x)	Обрахування квадратного кореня x
x^y	pow(x, y)	Возведення x до ступеня y
$\arccos(x)$	acos(x)	Обрахування значення арккосинуса x
$\text{arctg}(x)$	atan(x)	Обрахування значення арктангенса x
$\ln(x)$	log(x)	Обрахування натурального логарифму x
$\lg_{10}(x)$	log10(x)	Обрахування десяткового логарифму x

Приклад виконання завдання

Завдання: Написати програму для обчислення значення виразу при заданих вхідних даних та порівняти отриманий результат із правильним значенням

$$s = \frac{x^{2y} + e^{y-1}}{1 + x|y - \text{tg}z|} + \sqrt[3]{x} - \ln(z)$$

При $x = 2.45$, $y = -0.423 \times 10^{-2}$, $z = 1.232 \times 10^3$
 відповідь $s = 6.9465$.

Лістинг програми:

```
#include<stdio.h>
#include<math.h>
int main ()
{
float x,y,z,a,b,c,S;
printf( "\nVvedite x: "); scanf("%f", &x);
printf( "\nVvedite y: "); scanf("%f", &y);
printf( "\nVvedite z: "); scanf("%f", &z);
a= pow(x,2*y)+exp(y-1);
b= 1+x*fabs(y-tan(z));
c= 10*pow(x,1/3)-log(z);
S = a/b+c;
printf( "Result S=%.4f", S);

return 0;
}
```

Контрольні питання:

1. Які прості типи даних та їх модифікатори використовуються у мові програмування C?
2. За якими правилами відбувається неявне перетворення типів даних?
3. Як здійснюється форматоване введення та виведення даних?

Тема № 3.

Арифметичні операції. Декремент. Інкремент.

Мета: вивчити особливості використання операцій інкременту та декременту та їх форми.

Література:

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

Зміст роботи

Завдання. Обчислити значення виразу при різних типах даних (float і double). Обчислення слід виконувати з використанням проміжних змінних. Порівняти і пояснити отримані результати.

№	Завдання
1	$\frac{(a-b)^3 - (a^3 - 3a^2b)}{b^3 - 3ab^2}$, при a=1000, b=0.0001
2	$\frac{(a-b)^2 - (a^2 - 2ab)}{b^2}$, при a=1000, b=0.0001
3	$\frac{(a+b)^3 - (a^3 + 3a^2b)}{3ab^2 + b^3}$, при a=1000, b=0.0001
4	$\frac{(a+b)^3 - (a^3)}{3ab^2 + b^3 + 3a^2b}$, при a=1000, b=0.0001
5	$\frac{(a-b)^3 - (a^3 - 3a^2b)}{b^3 - 3ab^2}$, при a=1000, b=0.0001
6	$\frac{(a-b)^3 - (a^3 - 3ab^2)}{b^3 - 3a^2b}$, при a=1000, b=0.0001
7	$\frac{(a-b)^3 - (a^3)}{b^3 - 3ab^2 - 3a^2b}$, при a=1000, b=0.0001
8	$\frac{(a+b)^4 - (a^4 + 4a^3b + 6a^2b^2)}{4ab^3 + b^4}$, при a=100, b=0.001
9	$\frac{(a+b)^4 - (a^4 + 4a^3b)}{6a^2b^2 + 4ab^3 + b^4}$, при a=100, b=0.001

10	$\frac{(a-b)^4 - (a^4 - 4a^3b + 6a^2b^2)}{b^4 - 4ab^3}$, при a=100, b=0.001
11	$\frac{(a-b)^4 - (a^4 - 4a^3b)}{6a^2b^2 - 4ab^3 + b^4}$, при a=100, b=0.001
12	$\frac{(a+b)^2 - (a^2 + 2ab)}{b^2}$, при a=1000, b=0.0001
13	$\frac{(a-b)^2 - (a^2 - 2ab)}{b^2}$, при a=1000, b=0.0001
14	$\frac{(a+b)^3 - (a^3 + 3a^2b)}{3ab^2 + b^3}$, при a=1000, b=0.0001
15	$\frac{(a+b)^3 - (a^3)}{3ab^2 + b^3 + 3a^2b}$, при a=1000, b=0.0001

Приклади:

<pre>#include<stdio.h> int main() { int aa = 5; printf ("aa=%d\n", ++aa); // 6 aa=7; printf ("aa=%d\n", aa++); // 7 printf ("aa=%d\n", aa); // 8 return 0; }</pre>	<pre>#include<stdio.h> void main() { int m = 7, n = 4, yz; printf ("n =%d, m =%d \n", n, m); yz = ++ n + m; printf ("++ n * ++ m =%d \n", yz); printf ("n =%d, m =%d \n", n, m); yz = m++ <n; printf ("m ++ < n =%d \n", yz); printf ("n =% d, m =%d \n", n, m); }</pre>
---	--

Методичні рекомендації

Функції стандартного введення - виведення описані у файлі *stdio.h*.

printf() - форматне виведення на екран:

```
int printf(char *format, <список виводу >);
```

Перший параметр є символьним рядком, що задає специфікації формату, інші параметри - перерахування змінних і виразів, значення яких виводяться. Кожна специфікація формату має вид (параметри в квадратних дужках необов'язкові):

`%[flags][width][.prec]type`, де

<code>%</code>	символ %
<code>flags -</code>	ознака вирівнювання:
<code>+ чи порожньо</code>	вирівнювання по правому краї
<code>-</code>	вирівнювання по лівому краї
<code>width-</code>	ціле число - загальна ширина поля. Якщо це число починається з цифри 0, виведення доповнюється ліворуч нулями до заданої ширини. У задану ширину входять всі символи виведення, включаючи знак, дробову частину і т.п.
<code>type -</code>	тип специфікації
<code>d чи i</code>	ціле десяткове число зі знаком
<code>u</code>	десяткове число без знака
<code>x</code>	ціле 16-ричне число без знака
<code>f</code>	число з плаваючою точкою
<code>e</code>	число в E-формі
<code>g</code>	число з плаваючою крапкою або в E-формі
<code>c</code>	один символ
<code>s</code>	рядок
<code>prec -</code>	ціле число, кількість знаків після крапки при виведенні чисел з плаваючою точкою

Наприклад:

<code>%d</code>	друк десяткового цілого
<code>%6d</code>	друк десяткового цілого в поле з шести позицій
<code>%f</code>	друк числа з плаваючою крапкою
<code>%6f</code>	друк числа з плаваючою точкою в полі з шести позицій
<code>%.2f</code>	друк числа з плаваючою точкою з двома цифрами після десяткової точки
<code>%6.2f</code>	друк числа з плаваючою точкою і двома цифрами після десяткової точки в поле з шести позицій

Функція *modf()* розбиває число на цілу і дробову частину. Вона повертає дробову частину і поміщає цілу частину в змінну, на яку вказує.

Приклад:

```
float a=2.58, y,x;
y = modf(a,&x);
```

де y-дробова частина числа, а x- ціла частина числа.

В С є два незвичайних оператора, призначених для збільшення і зменшення змінних. Оператор інкременту <++> додає 1 до свого операнду, а оператор декременту <--> – віднімає

++ Збільшення на одиницю:

– префіксна операція збільшує операнд до його використання;

- постфіксна операція збільшує операнд після його використання.
- *Зменшення на одиницю:*
 - префіксна операція зменшує операнд до його використання,
 - постфіксна операція зменшує операнд після його використання.

Контрольні питання:

1. Як правильно записати вираз мовою програмування C?

$$a = x^3 + /x^2 - 13x + 52 / -11$$

2. Визначте значення кожної змінної після операції, якщо на початку операції всі змінні мають значення рівне 5:
- a. $p^* = x++$
 - b. $q /= ++x$
 - c. $w = (--x) + (w--)$
 - d. $k += ((--x)--)+10$?
3. За допомогою якої функції виводиться результат на екран.
4. Дані, яких типів ви б використовували для представлення наступних величин?
- a. Населення міста Житомир.
 - b. Середня вага картин Рембрандта.
5. З якої причини замість даних типу `int` використовуються дані типу `long`?
6. За допомогою якої функції можна зробити введення даних з клавіатури.
7. Визначте, що являють собою наступні керуючі послідовності:
- a. `\n`
 - b. `\t`
8. Визначте, якого типу наступні константи:
- a. `'\ b'`
 - b. 1066
 - c. 99.44
9. Назвіть пріоритети виконання арифметичних операцій.
10. Чому дорівнюють наступні вирази
- 10%2=?
 - 9%2=?
 - 10%3=?
 - 11%3=?
 - 3%11=?
11. Визначити чому буде дорівнювати `b`:
- ```
float a = 241.5;
int b = (int)a % 2.
```

**Тема № 4.**  
**Програмування розгалужених алгоритмів.**  
**Умовний оператор *if-else*.**

**Мета:** вивчити особливості використання умовного оператора.

**Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

**Зміст роботи**

**!!! Рішення супроводжується створенням блок-схеми.**

**Завдання.** Напишіть програму згідно Вашого варіанту.

| <b>Варіант</b> | <b>Завдання</b>                                                                                                                                                                                                                                                      |
|----------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1              | Визначити, чи можна з відрізків заданої довжини $x$ , $y$ , $z$ побудувати трикутник.                                                                                                                                                                                |
| 2              | Дано координати точки на площині. Якщо точка не лежить на координатних осях, то вивести 0. Якщо точка збігається з початком координат, то вивести 1. Якщо точка не збігається з початком координат, але лежить на осі $OX$ або $OY$ , то вивести відповідно 2 або 3. |
| 3              | Вести два числа $a$ і $b$ . Число, яке більше замінити потрійним добутком, число яке менше замінити напівсумою.                                                                                                                                                      |
| 4              | Якщо серед трьох чисел $a$ , $b$ та $c$ є хоча б одне парне, то знайти максимальне число, інакше - мінімальне.                                                                                                                                                       |
| 5              | Визначити, в якому квадранті знаходиться точка з координатами $(x, y)$ і вивести номер квадранта на екран.                                                                                                                                                           |
| 6              | Дано $x, y, z$ . Знайти: $\max\{x + y + z, xyz\} + 3$ ;                                                                                                                                                                                                              |
| 7              | Дано $x, y, z$ . Знайти: $\min\{(x + y) - 7, y + 2z\} - 4$ .                                                                                                                                                                                                         |
| 8              | Дано $x$ . Обрахувати $y$ , якщо: $y = \begin{cases} x^2, & \text{якщо } -2 \leq x \leq 2 \\ 4, & \text{якщо } x < -2 \text{ і } x > 2 \end{cases}$                                                                                                                  |

|    |                                                                                                                                                                                                        |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9  | Дано $x$ . Обрахувати $y$ , якщо:<br>$y = \begin{cases} 0, & \text{якщо } x \leq 0 \\ x, & \text{якщо } 0 < x \leq 1 \\ x^2, & \text{якщо } x > 1 \end{cases}$                                         |
| 10 | Дано $x$ . Обрахувати $y$ , якщо:<br>$y = \begin{cases} x^2 + 4x + 5, & \text{якщо } x \leq 2 \\ \frac{1}{x^2 + 4x + 5}, & \text{якщо } x > 2 \end{cases}$                                             |
| 11 | Дано $x$ . Обрахувати $y$ , якщо:<br>$y = \begin{cases} x^2 - x, & \text{якщо } 0 < x \leq 1 \\ 0, & \text{якщо } x \leq 0 \\ x^2 - \sin \pi x^2 - 1, & \text{якщо } x > 1 \end{cases}$                |
| 12 | Дано три числа, які є довжинами сторін трикутника. Визначити тип трикутника (рівносторонній, рівнобедрений, різносторонній).                                                                           |
| 13 | Дано ціле число, яке лежить в діапазоні від -999 до 999. Вивести рядок - словесний опис даного числа наступного вигляду "негативне двозначне число", "число = 0", "позитивне однозначне число" і т. п. |
| 14 | Дано $x, y, z$ . Знайти: $\min \{x^2 + y^2, y^2 + z^2\} / 4$ .                                                                                                                                         |
| 15 | Визначте, чи потрапляє точка з координатами $(x, y)$ в коло радіусом $R$ . Якщо точка потрапляє в коло, вивести на екран одиницю, в іншому випадку - нуль.                                             |

### Методичні рекомендації

У лінійних програмах всі команди виконуються послідовно, але якщо необхідно виконати вибір одного або іншого варіанту рішення використовуються розгалужені алгоритми.

Оператор *if-else* застосовується для вибору із двох варіантів;

Оператор *switch* застосовується для вибору із декількох варіантів.

За допомогою вкладених одна в іншу конструкцій *if* можна реалізувати вибір декількох варіантів.

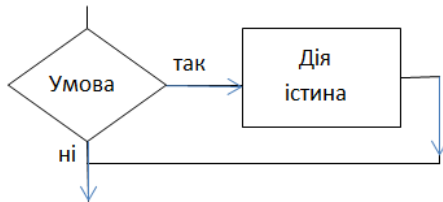


Синтаксис оператора :

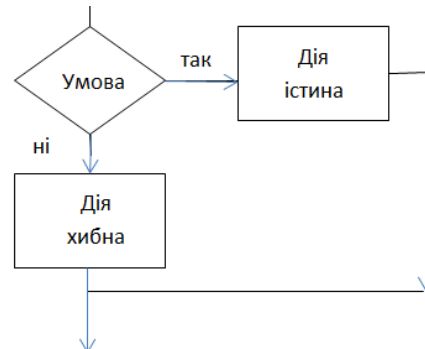
```
if (<умова><оператор1>;
[else <оператор2;>]
```

Графічне позначення:

Не повна форма:



Повна форма:



**Приклади:**

|                                                                                                                                                                                                                                                    |                                                                                                                                                                                                                                                              |                                                                                                                                                                                                                                 |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre>#include&lt;stdio.h&gt; #include&lt;locale.h&gt; void main() { setlocale (LC_ALL, "Rus"); float A, B, Max; printf("Введіть А и В:\n "); scanf ( "%f%f", &amp;A, &amp;B ); Max = A; if ( B &gt; A ) Max = B; printf("max= %.3f", Max); }</pre> | <pre>#include&lt;stdio.h&gt; #include&lt;locale.h&gt; void main() { setlocale (LC_ALL, "Rus"); float A, B; printf ("Введіть А и В:\n"); scanf ( "%f%f", &amp;A, &amp;B ); if ( A &gt; B ) printf ("max=%.3f\n", A); else printf ( "max= %.3f\n", B); }</pre> | <pre>#include&lt;stdio.h&gt; #include&lt;locale.h&gt; void main() { setlocale (LC_ALL, "Rus"); float A, B, Max; printf("Введіть А и В :\n"); scanf ( "%f%f", &amp;A, &amp;B ); printf ( "\nmax= %.3f\n", (A&gt;B)?A:B); }</pre> |
|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

**Контрольні питання:**

1. У яких випадках слід використовувати оператор if.
2. Як записується логічна рівність в операторі if
3. Якими символами позначаються логічні операції ТА, АБО в умовному операторі?
4. Запишіть умовний оператор, для визначення знаку змінної.
5. Використовуючи умовний оператор, виконайте перевірку на належність значення змінної діапазону [10; 20).
6. Наведіть позначення логічного знаку "не дорівнює".

**Тема № 5.**  
**Програмування розгалужених алгоритмів**  
**Оператор вибору switch.**

**Мета:** вивчити особливості використання оператора вибору switch.

**Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

**Зміст роботи**

**!!! Рішення супроводжується створенням блок-схеми.**

**Завдання.** Написати програму з використанням оператора switch.

| <b>Варіант</b> | <b>Завдання</b>                                                                                                                                                                                       |
|----------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1              | Дано ціле число у діапазоні [1, 7]. Вивести на екран повідомлення про день тижня. Наприклад: 1- понеділок...                                                                                          |
| 2              | Арифметичні дії над числами пронумеровані наступним чином: 1- "+", 2- "-", 3- "*", 4- "/". Дано: номер дії та два числа з плаваючою комою А і В. Виконати дії над числами та вивести результат.       |
| 3              | Для заданих значень радіусів 10, 20, 40, 80 підрахувати площу круга та вивести на екран відповідь.                                                                                                    |
| 4              | Дано ціле число у діапазоні [1, 12]. Вивести на екран повідомлення про відповідний місяць. Наприклад; 1- січень...                                                                                    |
| 5              | Одиниці довжини пронумеровані наступним чином: 1-сантиметр, 2-дециметр, 3- міліметр, 4-кілометр.Перевести задану довжину відрізка в метрах у відповідну довжину, наприклад: 1 метр = 100 сантиметрів. |
| 6              | Скласти програму, яка в залежності від порядкового номера дня тижня (1..7) виводить на екран його назва (понеділок, вівторок ..., неділя)                                                             |
| 7              | Написати програму-абетку, яка на введену букву (малу або велику) виводить на екран відповідне повідомлення. Наприклад: вводимо 'а' або 'А' – антилопа.                                                |

|    |                                                                                                                                                                                                               |
|----|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 8  | Написати програму, яка обчислює вартість 10 хвилинної міжміської розмови, в залежності від коду міста. (Київ (44) – 34,15 грн., Житомир (41) – 24,98 грн., Харків (57) – 32,69 грн., Львів (37) – 35,00 грн.) |
| 9  | Одиниці маси пронумеровані наступним чином: 1– міліграм, 2–грам, 3 – тонна, 4 – центнер. Задану масу у кілограмах перевести в іншу, наприклад у грами.                                                        |
| 10 | Дано ціле число у діапазоні [1, 12]. Вивести на екран повідомлення про пору року. Наприклад; 3- весна...                                                                                                      |
| 11 | Дано ціле число, що означає опис оцінки. Наприклад: 4- добре, 3- задовільно...                                                                                                                                |
| 12 | Дано ціле число у діапазоні [1, 12]. Визначити кількість днів у відповідному місяці. Наприклад; 1- "у січні 31 день"...                                                                                       |
| 13 | Дано ціле число у діапазоні [1, 12], що означає місяць. Вивести на екран повідомлення про квартал. Наприклад; 3-перший квартал...                                                                             |
| 14 | Дано ціле число у діапазоні [0, 9]. Вивести на екран число прописом. Наприклад: 7- сім...                                                                                                                     |
| 15 | Написати програму заміни символу, що вводиться, від 0 до 9 відповідним повідомленням. Наприклад: '7' – введено число 7...                                                                                     |

### Методичні рекомендації

Умовний оператор `if` полегшує написання програм, в яких необхідно проводити вибір між невеликим числом можливих варіантів. Однак іноді в програмі необхідно здійснити вибір одного варіанта з багатьох можливих. Формально для цього можна скористатися вкладеною конструкцією, однак у багатьох випадках виявляється більш зручним застосовувати оператор `switch`.

Синтаксис:

```

switch (<змінна>)
{
 case значення1:
 Виконується, якщо < змінна > == значення1
 break;
 case значення2:
 Виконується, якщо < змінна > == значення2
 break;
 ...
 default:
 Виконується, якщо жоден варіант не підійшов
 break;
}

```

Відмітимо, що в тілі оператора switch можна використовувати вкладений оператор switch. Команда break забезпечує переривання виконання самого внутрішнього із операторів switch, do, for, while. Після виконання команди break управління передається наступному оператору.

### Приклади:

|                                                                                                                                                                                                                             |                                                                                                                                                                                  |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <pre> int i; scanf("%d",&amp;i); switch (i) { case 1: { i += 1;         printf("%d\n",i); break;} case 2: { i *= 2;         printf("%d\n",i); break;} case 3: { i -=5;         printf("%d\n",i); break;} default:; } </pre> | <pre> char ZNAC; int x, y=4,z=2; scanf("%c",&amp;ZNAC); switch (ZNAC) { case '+':x = y + z;break; case '-':x = y - z;break; case '*':x = y * z;break;         default:; } </pre> |
|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

### Контрольні питання:

1. У яких випадках слід використовувати оператор switch?
2. Чи є випадки, коли слід опустити оператор break?

**Тема № 6.**  
**Програмування розгалужених алгоритмів.**  
**Оператори циклу *for* і *while*.**

**Мета:** вивчити особливості використання операторів циклу.

**Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

**Зміст роботи**

**!!! Завдання виконуються у двох варіантах з оператором циклу *for* та *while*.**

**Завдання.** Обчислити значення функції  $F(x)$  на відрізку  $[a,b]$  кроком  $h$ .

| №  | $F(x)$                | $[a,b]$ | $h$ | $F(x)$                      | $[a,b]$ | $h$ |
|----|-----------------------|---------|-----|-----------------------------|---------|-----|
| 1  | $e^{-x}\sqrt{x}$      | 1;3     | 1   | $\ln(x)$                    | 1;1.5   | 0.1 |
| 2  | $\sqrt[3]{x}$         | 3;4     | 1   | $1+\ln^2(x)$                | 0.4;1   | 0.1 |
| 3  | $x\sqrt{x}$           | 4;5     | 1   | $1+e^x$                     | 2;3.4   | 0.2 |
| 4  | $1/\sqrt[3]{x}$       | 5;7     | 1   | $e^{x^2}/2$                 | 2;3     | 0.1 |
| 5  | $1\sqrt{x}$           | 6;8     | 1   | $\cos(x)e^{-x}$             | 1;2.2   | 0.2 |
| 6  | $(2x+1)/\sqrt{x}$     | 8;9     | 1   | $1/(1+e^{-x})$              | 3;4.5   | 0.1 |
| 7  | $\sin(x)\cos(x^2)$    | 1;5     | 1   | $\sqrt[4]{x}$               | 0.5;1   | 0.1 |
| 8  | $0.5+\sin^2(x)$       | 2;3     | 1   | $\sqrt{x^2+1}$              | 2.5;1   | 0.1 |
| 9  | $\sqrt{x}\cos(x)$     | 3;4     | 1   | $1+\sin(x^2)$               | 1;2.4   | 0.2 |
| 10 | $(2x+1)*\sqrt{x}$     | 7;8     | 1   | $\sqrt[5]{x}\cos(x)$        | 1;1.5   | 0.1 |
| 11 | $(1+x)/\sqrt[5]{x}$   | 5;7     | 1   | $e^{x^2}/x^2$               | 2;3     | 0.1 |
| 12 | $e^{-x}\sqrt{1+x}$    | 1;9     | 3   | $(x^2-1)+\sqrt[3]{x^2}$     | 2.5;5   | 0.5 |
| 13 | $\sin(x^2)+\cos(x^2)$ | 1;5     | 1   | $\sqrt[5]{x}+\sqrt[3]{x^2}$ | 0.5;1   | 0.1 |
| 14 | $\sin(x)/(1+\cos(x))$ | 5;9     | 2   | $\sin(x^2)-e^x$             | 1;2.4   | 0.2 |
| 15 | $e^{4x}\sqrt{x}-4x$   | 1;3     | 1   | $\sqrt[4]{x}\sin(x)$        | 1;4.5   | 0.5 |

## Методичні рекомендації

Цикл – це багаторазово повторювана частина алгоритму. Цикл, що не містить в собі інших циклів, називають простим. Якщо він містить всередині себе інші цикли або розгалуження, то цикл називають складним або вкладеним. Будь-який цикл характеризується однією або декількома змінними, так званими параметрами циклу, від аналізу значень яких залежить виконання циклу.

Спосіб організації циклу залежить від умови задачі. У багатьох задачах кількість повторень циклу вказується, це так звані цикли з відомою кількістю повторень або цикли з лічильником.

Існує три види циклів: `while`, `for` і `do while`.

Синтаксис циклу з параметрами:

```
for(<ініціалізація лічильника>; <умова>; <зміна значення лічильника>)
{
 <Тіло циклу>
}
```

Оператор циклу `while` називається циклом з передумовою та має такий синтаксис:

```
while (<вираз(умова)>)
{
 <Тіло циклу>
}
```

Тіло циклу виконується до тих пір, поки значення виразу(умови) рівне `true`.

### Приклади:

|                                                                                                                                                               |                                                                                                            |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------|
| <pre>int k, i, j=1; printf ("Укажіть число повторень циклу: "); scanf ("%d", &amp;k); for (i = 1; i &lt;= k; i++) {     j *= i;     printf("%6d", j); }</pre> | <pre>int n, y; for (int k = 0, n = 10; k &lt;= n; k++, n--) {     y = k * n;     printf("%3d", y); }</pre> |
| <pre>int j=1; i=1; printf ("Укажіть число повторень циклу: "); scanf ("%d", &amp;k); while(i &lt;= k) {     j *= i; i++;     printf("%6d", j); }</pre>        | <pre>int k=0, n=10, y; while(k &lt;= n) {     y = k * n; k++; n--;     printf("%3d", y); }</pre>           |

### Контрольні питання:

1. Для яких цілей використовуються цикли в програмуванні?
2. Перерахуйте оператори циклів в мові C.
3. Що таке цикл з передумовою?
4. Опишіть синтаксис оператору `for`.
5. Опишіть синтаксис оператору `while`.
6. У чому полягають відмінності між операторами `for` і `while`.
7. Які цикли називаються вкладеними?

**Тема № 7.**  
**Програмування розгалужених алгоритмів**  
**Оператори циклу for, while, do-while.**

**Мета:** вивчити особливості використання операторів циклу.

**Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

**Зміст роботи**

**!!! Завдання лабораторної роботи оформлюється меню:  
 "1 - продовжити роботу, 2 – завершити роботу."**

**Завдання.** Написати програму з використанням операторів циклу while або do-while. Послідовність чисел вводиться, поки користувач не введе 0.

**Рекомендований вид екрану (варіант 1):**

|                                                                                                                                                                                                                     |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Обчислення середнього арифметичного послідовності позитивних чисел.<br>Для завершення введення введіть нуль.<br>-> 45<br>-> 23<br>-> 15<br>-> 0<br>Введено чисел: 3<br>Сума чисел: 83<br>Середнє арифметичне: 27.67 |
|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|

|   |                                                                                                                            |
|---|----------------------------------------------------------------------------------------------------------------------------|
| 1 | Написати програму, яка обчислює суму і середнє арифметичне послідовності позитивних чисел, які вводяться з клавіатури.     |
| 2 | Написати програму, яка обчислює добуток і суму послідовності чисел, які вводяться з клавіатури.                            |
| 3 | Написати програму, яка обчислює кількість позитивних і негативних чисел послідовності чисел, які вводяться з клавіатури.   |
| 4 | Написати програму, яка визначає максимальне число і середнє арифметичне у введеної з клавіатури послідовності чисел.       |
| 5 | Написати програму, яка визначає максимальне число серед негативних у введеної з клавіатури послідовності чисел.            |
| 6 | Написати програму, яка визначає мінімальне число та суму всіх чисел у введеної з клавіатури послідовності позитивних чисел |

|    |                                                                                                                          |
|----|--------------------------------------------------------------------------------------------------------------------------|
| 7  | Написати програму, яка визначає <i>мінімальне число серед позитивних</i> у введеної з клавіатури послідовності чисел     |
| 8  | Написати програму, яка визначає <i>кількість простих чисел</i> у введеної з клавіатури послідовності цілих чисел         |
| 9  | Написати програму, яка визначає <i>добуток непарних чисел</i> у введеної з клавіатури послідовності цілих чисел          |
| 10 | Написати програму, яка визначає <i>добуток парних чисел</i> у введеної з клавіатури послідовності цілих чисел            |
| 11 | Написати програму, яка визначає <i>суму непарних чисел</i> у введеної з клавіатури послідовності цілих чисел             |
| 12 | Написати програму, яка визначає <i>суму парних чисел</i> у введеної з клавіатури послідовності цілих чисел               |
| 13 | Написати програму, яка визначає <i>знак числа</i> у введеної з клавіатури послідовності цілих чисел                      |
| 14 | Написати програму, яка визначає <i>середнє арифметичне позитивних чисел</i> у введеної з клавіатури послідовності чисел. |
| 15 | Написати програму, яка визначає <i>середнє арифметичне негативних чисел</i> у введеної з клавіатури послідовності чисел. |

### Методичні рекомендації

Оператор *do-while* виконує оператор або складений оператор доти, поки задана умова істинна. Виконання циклу припиняється, коли умова стає хибною. Синтаксис:

```
do
{
 <Тіло циклу>
}
while (<вираз(умова)>);
```

Вираз в циклі *do-while* обчислюється *після* виконання тіла циклу. Таким чином, тіло циклу *завжди виконується хоча б один раз*.

Вираз повинен бути арифметичного типу. Виконання відбувається наступним чином:

1. Виконується тіло циклу (оператор).
2. Потім обчислюється значення виразу . Якщо вираз хибний, виконання оператора *do-while* переривається і керування передається наступному оператору програми. Якщо вираз істинний (ненульовий ), процес повторюється, починаючи з кроку 1.



Виконання оператора *do-while* також переривається, якщо оператори *break* або *return* виконуються в тілі циклу.

**Контрольні питання:**

1. Для яких цілей використовуються цикли в програмуванні?
2. Перерахуйте оператори циклів в мові С.
3. Які особливості має оператор циклу *for*?
4. Які особливості має оператор циклу *while*?
5. Які особливості має оператор циклу *do-while*?
6. Для чого використовується оператор *break*?
7. Для чого використовується оператор *continue*?
8. Охарактеризуйте поняття – вкладений цикл?

## Тема № 8. Логічні операції та вирази.

**Мета:** набути навичок у створенні логічних виразів та використанні логічних операцій.

### **Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

**Завдання.** Написати програму з використанням тернарної операції:

|    |                                                                                                                                                                                      |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1  | Ввести з клавіатури три дійсних значення і вивести на екран максимальне з цих значень:                                                                                               |
| 2  | Квадратне рівняння задається коефіцієнтами А, В і С. Визначити скільки коренів має дане рівняння.                                                                                    |
| 3  | Написати програму обчислення $1/a$ з перевіркою $a \neq 0$                                                                                                                           |
| 4  | Ввести два числа а і b. Більше число зменшити на 2, менше - збільшити у 2 рази.                                                                                                      |
| 5  | Написати програму перевірки невходження введеного числа у діапазон від 0 до 5                                                                                                        |
| 6  | Ввести з клавіатури число, і перевірити його, парне воно чи ні.                                                                                                                      |
| 7  | З клавіатури вводяться два числа – кількість забитих голів командами "Динамо" і "Шахтар" у футбольному матчі. Вивести на екран результат гри – перемога "Динамо" / "Шахтар" / нічия. |
| 8  | Напишіть програму, яка буде перевіряти число на кратність 3-м і 7-ми (це числа 21, 42, 63 ...). Вивести на екран відповідне повідомлення.                                            |
| 9  | Обрахувати значення функції<br>$y(x) = \begin{cases} x + 1, & x > 0 \\ x^2, & x \leq 0 \end{cases}$                                                                                  |
| 10 | Написати програму перевірки попадання введеного числа в діапазон від -2 до 4.                                                                                                        |
| 11 | Дано $x, y, z$ . Знайти: $\max\{x + y + z, xyz\} + 3$ ;                                                                                                                              |
| 12 | Дано $x, y, z$ . Знайти: $\min\{(x + y), y + z\} * x$ .                                                                                                                              |
| 13 | Дано $x, y, z$ . Знайти: $\min\{(x + y) - 7, y + 2z\} - 4$ .                                                                                                                         |
| 14 | Обрахувати значення функції<br>$y(x) = \begin{cases} x + 1, & x > 5 \\ x + 4, & x \leq 5 \end{cases}$                                                                                |
| 15 | Написати програму порівняння введеного числа зі значеннями 0,4,8,9 і 30                                                                                                              |

## Методичні рекомендації

Оператор, `if` дозволяє реалізовувати складні умовні переходи. У мові C є три логічні операції:

`&&` - Логічне І

`||` - Логічне АБО

`!` - Логічне НІ

На основі цих трьох логічних операцій можна сформуванати більш складні умови. Наприклад, якщо є три змінні `exp1`, `exp2` і `exp3`, то вони можуть становити логічні конструкції, представлені в таблиці:

|                                                            |                                                                                                                                                                                                               |
|------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>if(exp1 &gt; exp2 &amp;&amp; exp2 &lt; exp3 )</code> | істинно, якщо значення змінної <code>exp1</code> більше значення змінної <code>exp2</code> і значення змінної <code>exp2</code> менше значення змінної <code>exp3</code> .                                    |
| <code>if(exp1 &lt;= exp2    exp1 &gt;= exp3 )</code>       | істинно, якщо значення змінної <code>exp1</code> менше <b>або</b> дорівнює значенню змінної <code>exp2</code> або значення змінної <code>exp1</code> більше або дорівнює значенню змінної <code>exp3</code> . |
| <code>if(exp1 &amp;&amp; exp2 &amp;&amp; !exp3)</code>     | істинно, якщо істинно значення <code>exp 1</code> і <code>exp 2</code> і хибно значення <code>exp3</code>                                                                                                     |
| <code>if(!exp1    !exp2 &amp;&amp; exp3)</code>            | істинно, якщо хибно значення <code>exp 1</code> або <code>exp 2</code> і істинно значення <code>exp3</code> .                                                                                                 |

Подібно операціям множення і складання в математиці, логічні операції І АБО НІ, також мають свої пріоритети. Найвищий пріоритет має операція НІ, тобто така операція виконується в першу чергу. Більш низький пріоритет у операції І, і нарешті найменший пріоритет у операції АБО.

Оператор "кома" використовується для зв'язки декількох виразів. Ліва сторона оператора "кома" завжди обчислюється як `void` (тобто не повертає значення). Це означає, що значення виразу, що знаходиться з правого боку, стане значенням розділеного комами виразу.

Наприклад:

```
x = (y = 3, y + 1); // x=4
```

### Контрольні питання:

1. Для чого використовують логічні вирази?
2. Що таке простий логічний вираз?
3. Що таке складений логічний вираз?
4. Які є логічні операції?
5. Дайте означення логічної операції **not, and, or**.
6. Який пріоритет логічних операцій?
7. Назвіть ситуацію, де не можна використовувати, `if{...} else{...}`, але можна тернарний оператор?

## Тема № 9. Вкладені цикли.

**Мета:** набуття навичок роботи з різними типами операторів циклу.

### Література:

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

**Завдання.** Написати програму з використанням операторів циклів while і for.

|    |                                                                                                                  |
|----|------------------------------------------------------------------------------------------------------------------|
| 1  | Обрахувати $\sum_{i=1}^{50} \sum_{j=1}^{30} i + j$ за допомогою вкладених циклів for                             |
| 2  | Обрахувати $S = \sum_{i=1}^{\infty} i$ поки $S < 50$ за допомогою циклу for                                      |
| 3  | Обрахувати $\sum_{i=1}^{50} \sum_{j=1}^{10} 1/(i + j)$ за допомогою вкладених циклів while                       |
| 4  | Обрахувати $\sum_{i=-10}^{10} \frac{1}{i^3}, i \neq 0$ за допомогою циклу for                                    |
| 5  | Обрахувати $\sum_{i=-10}^{20} \sum_{j=0}^{10} \frac{1}{(i+j)^2}, i + j \neq 0$ за допомогою вкладених циклів for |
| 6  | Обрахувати $S = \sum_{i=1}^{\infty} i^2$ поки $S < 100$ за допомогою циклу for                                   |
| 7  | Визначити порядковий номер елемента послідовності $2^1, 2^2, 2^3, \dots, 2^n$ , значення якого перевищить 100    |
| 8  | Обрахувати $\sum_{i=1}^n \sum_{j=1}^m \frac{\sin(j)}{2i+1}$ за допомогою вкладених циклів                        |
| 9  | Обрахувати $\sum_{i=1}^n \sum_{j=1}^m \frac{i^2}{2j+1}$ за допомогою вкладених циклів                            |
| 10 | Обрахувати $\sum_{i=1}^{10} \sum_{j=1}^7 \frac{i+j}{2j}$ за допомогою вкладених циклів                           |
| 11 | Обрахувати $\sum_{i=1}^{10} \prod_{j=1}^{10} \frac{i+j}{2j}$ за допомогою вкладених циклів                       |
| 12 | Обрахувати $\sum_{i=1}^m \prod_{j=1}^n \frac{i^2+j}{2}$ за допомогою вкладених циклів                            |
| 13 | Обрахувати $\sum_{i=1}^{50} \sum_{j=1}^{10} 1/\cos(i + j)$ за допомогою вкладених циклів                         |
| 14 | Обрахувати $\sum_{i=1}^{10} \sum_{j=1}^{10} \frac{\cos(i+1)}{j}$ за допомогою вкладених циклів                   |
| 15 | Обрахувати $\sum_{i=1}^n \prod_{j=1}^m \frac{\sin(i+j)}{2}$ за допомогою вкладених циклів                        |

### Контрольні питання:

1. Що таке цикли, які бувають цикли?
2. Який з циклів може не виконуватись жодного разу?
3. Який синтаксис команди "?" ?
4. Яка команда здійснює достроковий вихід з циклу?

## Тема № 10. Побітові операції.

**Мета:** вивчити особливості подання даних в цифровому вигляді; отримати практичні навички перетворення і обробки чисел в різних системах числення.

### **Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

**Завдання.** Написати програму по роботі з порозрядними операціями відповідно до свого варіанту.

**Дано:** unsigned char a, b, c, d, e, f;

**Знайти:**  $c = a \& b$ ;  $d = a | b$ ;  $e = a \wedge b$ ;  $f = \sim a$ ;

|   |            |    |             |    |            |
|---|------------|----|-------------|----|------------|
| 1 | a=5; b=6;  | 6  | a=4; b=5;   | 11 | a=6; b=3;  |
| 2 | a=7; b=15; | 7  | a=17; b=10; | 12 | a=7; b=5;  |
| 3 | a=6; b=7;  | 8  | a=27; b=15; | 13 | a=8; b=5;  |
| 4 | a=7; b=8;  | 9  | a=15; b=9;  | 14 | a=6; b=8;  |
| 5 | a=8; b=9;  | 10 | a=7; b=21;  | 15 | a=3; b=12; |

Для всіх варіантів знайти:  $b = a \gg 1$ ,  $c = d \gg 2$ ,  $d = e \ll 3$ ,  $e = b \ll 5$

Перевірте результат виконання програми, наприклад

| a=3   | b=21  | a & b | a   b | a ^ b | ~ a   |
|-------|-------|-------|-------|-------|-------|
| 00011 | 10101 | 00001 | 10111 | 10110 | 11100 |

### Методичні рекомендації

Мова C включає побітові операції, операції інкрементування й декрементування, умовну операцію, операцію кома, операції комбінованого присвоєння.

Ці операції використовуються в тих випадках, коли необхідно дістати доступ до окремих біт даних (при виведенні графічних зображень на екран). Побітові операції застосовуються тільки до цілочисельних значень. На відміну від логічних операцій, з їх допомогою порівнюються не два числа цілком, а окремі їх біти. Основні побітові операції: І (&), АБО (|) і що виключає АБО (^). Сюди можна також зарахувати унарну операцію побітового заперечення (~), яка інвертує значення бітів числа.

**Операція &** записує в біт результату одиницю тільки в тому випадку, якщо обидва порівнюваних біта дорівнюють 1, як показано в такій таблиці.

Ця операція часто використовується для маскуванню окремих бітів числа. Наприклад:  $0xF1 \& 0x35 = 0x31$ .

**Операція |** записує в біт результат одиницю в тому випадку, якщо хоч би один з порівнюваних бітів дорівнює 1, як показано в такій таблиці.

Ця операція часто застосовується для установки окремих бітів числа. Наприклад:  $0xF1 | 0x35 = 0xF5$ .

**Операція ^** записує в біт результату одиницю в тому випадку, якщо порівнювані біти відрізняються один від одного, як показано в такій таблиці.

Ця операція часто застосовується при виведенні зображень на екран, коли відбувається накладення декількох графічних шарів.

Наприклад:  $0xF1 \wedge 0x35 = 0xC4$ .

| Біт 0 | Біт 1 | Операція & | Операція | Операція ^ |
|-------|-------|------------|----------|------------|
| 0     | 0     | 0          | 0        | 0          |
| 0     | 1     | 0          | 1        | 1          |
| 1     | 0     | 0          | 1        | 1          |
| 1     | 1     | 1          | 1        | 0          |

У C існує дві операції зсуву:  $\ll$  - зсув ліворуч,  $\gg$  - зсув праворуч. Дія першої операції полягає в зсуві бітового представлення цілочисельної змінної, вказаної зліва від операції, ліворуч на кількість бітів, задану праворуч від операції. При цьому звільнені молодші біти заповнюються нулями, а відповідна кількість старших бітів втрачається.

Зсув беззнакового числа на одну позицію ліворуч із заповненням молодшого розряду нулем еквівалентне множенню числа на 2. Наприклад:

```
unsigned int iv=65; // молодший байт: 01000001
iv<<=1; // молодший байт: 10000010
cout<<iv; // буде виведене 130
```

Зсув праворуч супроводжується аналогічними діями, тільки бітове представлення числа зрушується на вказану кількість бітів управо. Значення молодших бітів втрачаються, а старші біти, що звільнилися, заповнюються нулями, якщо операнд беззнаковий, і значенням знакового біта інакше. Таким чином, зсув беззнакового числа на одну позицію праворуч еквівалентне діленню числа на два:

```
unsigned int iv=10; // молодший байт: 00001010
iv>>=1; // молодший байт: 00000101
cout<<iv; // буде виведене 5
```

### Контрольні питання:

1. Дайте поняття бітового поля.
2. За допомогою якої поразрядної операції здійснюється виділення бітів змінних?

3. Яка поразрядное операція дозволяє перемикає біти змінної?
4. Для яких цілей слід використовувати бітові поля?
5. Поясніть сенс поразрядної операції "логічне АБО".
6. За допомогою якої поразрядної операції здійснюється розподіл змінної на 2?

## Тема № 11. Одномірні масиви.

**Мета:** дослідити функції генерування послідовностей випадкових чисел, ознайомитись з можливостями створення одновимірних масивів і навчитись обробляти елементи масивів.

### **Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

**Завдання.** Розробити програму, дотримуючись таких вимог:

- використовувати статичні масиви;
- число  $n$  (кількість елементів масиву);
- елементи масиву - випадкові числа, згенеровані на інтервалі  $[a, b]$ , де  $a$  і  $b$  вводяться з клавіатури ( $a < b$ );
- усі вхідні дані а також елементи масиву виводяться на екран.

|   |                                                                                                                                                                                                                              |
|---|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1 | В одновимірному масиві, що складається з $n$ дійсних елементів, обчислити:<br>1) суму від'ємних елементів масиву;<br>2) добуток елементів масиву, розташованих між максимальним і мінімальним елементами.                    |
| 2 | В одновимірному масиві, що складається з $n$ цілих елементів, обчислити:<br>1) суму додатних елементів масиву;<br>2) добуток елементів масиву, розташованих між максимальним за модулем і мінімальним за модулем елементами. |
| 3 | В одновимірному масиві, що складається з $n$ дійсних елементів, обчислити:<br>1) добуток елементів масиву з парними номерами;<br>2) суму елементів масиву, розташованих між першим і останнім нульовими елементами.          |
| 4 | В одновимірному масиві, що складається з $n$ цілих елементів, обчислити:<br>1) суму елементів масиву з непарними номерами;<br>2) суму елементів масиву, розташованих між першим і останнім від'ємними елементами.            |
| 5 | В одновимірному масиві, що складається з $n$ дійсних елементів, обчислити:<br>1) максимальний елемент масиву;<br>2) суму елементів масиву, розташованих до останнього додатного елемента.                                    |



|    |                                                                                                                                                                                                                                                                                                                                                                   |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 6  | <p>В одновимірному масиві, що складається з <math>n</math> цілих елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) мінімальний елемент масиву;</li> <li>2) суму елементів масиву, розташованих між першим і останнім додатними елементами.</li> </ol>                                                                                           |
| 7  | <p>В одновимірному масиві, що складається з <math>n</math> дійсних елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) номер максимального елемента масиву;</li> <li>2) добуток елементів масиву, розташованих між першим і другим нульовими елементами.</li> </ol>                                                                               |
| 8  | <p>В одновимірному масиві, що складається з <math>n</math> цілих елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) номер мінімального елемента масиву;</li> <li>2) суму елементів масиву, розташованих між першим і другим від'ємними елементами.</li> </ol>                                                                                    |
| 9  | <p>В одновимірному масиві, що складається з <math>n</math> дійсних елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) максимальний за модулем елемент масиву;</li> <li>2) суму елементів масиву, розташованих між першим і другим додатними елементами.</li> </ol>                                                                               |
| 10 | <p>В одновимірному масиві, що складається з <math>n</math> цілих елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) мінімальний за модулем елемент масиву;</li> <li>2) суму модулів елементів, розташованих після першого елемента, рівного нулю.</li> </ol>                                                                                     |
| 11 | <p>В одновимірному масиві, що складається з <math>n</math> дійсних елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) номер мінімального за модулем елемента масиву;</li> <li>2) суму модулів елементів, розташованих після першого від'ємного елемента.</li> </ol>                                                                              |
| 12 | <p>В одновимірному масиві, що складається з <math>n</math> цілих елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) номер максимального за модулем елемента масиву;</li> <li>2) суму елементів масиву, розташованих після першого додатного елемента.</li> </ol>                                                                                 |
| 13 | <p>В одновимірному масиві, що складається з <math>n</math> дійсних елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) кількість елементів, що лежать в діапазоні від <math>A</math> до <math>B</math> (<math>A</math> і <math>B</math> – з клавіатури);</li> <li>2) суму елементів масиву, розташованих після максимального елемента.</li> </ol> |

|    |                                                                                                                                                                                                                                                                                                                                              |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 14 | <p>В одновимірному масиві, що складається з <math>n</math> дійсних елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) кількість елементів масиву, рівних 0;</li> <li>2) суму елементів масиву, розташованих після мінімального елемента.</li> </ol>                                                                         |
| 15 | <p>В одновимірному масиві, що складається з <math>n</math> дійсних елементів, обчислити:</p> <ol style="list-style-type: none"> <li>1) кількість елементів масиву, більших за <math>C</math> (<math>C</math> вводиться з клавіатури);</li> <li>2) добуток елементів масиву, розташованих після максимального за модулем елемента.</li> </ol> |

### Методичні рекомендації

В сучасних умовах для отримання випадкових чисел використовують різноманітні генератори, які поділяються на апаратні та програмні.

Програмний генератор випадкових чисел являє собою програму, яка генерує послідовність чисел за деяким алгоритмом. Завдяки алгоритму така послідовність чисел цілком детермінована, тобто не може бути цілком випадковою. Її називають послідовністю псевдовипадкових чисел.

Для підключення вбудованого генератора псевдовипадкових послідовностей чисел в мові C використовують бібліотеку `<stdlib.h>`.

Функція `rand(): int rand(void);`

Якщо функція `srand()` викликається з одним і тим же початковим значенням, то послідовність псевдовипадкових чисел не зміниться. Тому в якості початкового заповнення зазвичай у програмах використовують системний час, який отримують за допомогою функції `time()` бібліотеки `<time.h>`.

#### Приклад 1.

```
int iS;
srand(time(NULL)); /* ініціалізація початкового заповнення */
iS = rand() % 10 + 1; /* Генерація випадкового числа: */
printf("%d, ", iS);
```

Масив – це набір змінних одного типу, що мають одне і те ж ім'я. Доступ до конкретного елемента масиву здійснюється за допомогою індексу. У мові C всі масиви розташовуються в окремій безперервній області пам'яті. Перший елемент масиву розташовується з самою найменшою адресою, а останній – з найбільшою. Масиви можуть бути одновимірними і багатовимірними. Рядок – це масив символічних змінних, що закінчується спеціальним нульовим символом, це найбільш поширений тип масиву.

Загальна форма оголошення одновимірного масиву має вигляд:

***тип ім'я\_змінної [розмір];***

Як і інші змінні, масив повинен бути оголошений явно, щоб компілятор виділив для нього певну область пам'яті (тобто розмістив масив). Тут *тип* позначає базовий тип масиву, що є типом кожного елемента. Розмір задає кількість елементів масиву.

Наприклад, наступний оператор оголошує масив із 25 елементів типу float:

```
float mas [25];
```

Доступ до елемента масиву здійснюється за допомогою імені масиву і індексу, який знаходиться в квадратних дужках, наприклад:

```
mas [20] = 15.47;
```

Індекс першого елемента будь-якого масиву в мові C дорівнює нулю. Тому оператор `int num[11]`, оголошує масив символів з 11 елементів – від `num[0]` до `num[10]`.

Обсяг пам'яті, необхідний для зберігання масиву, безпосередньо визначається її типом і розміром. Для одновимірного масиву кількість байтів пам'яті обчислюється таким чином:

*Кількість\_байтів = sizeof(базовий\_тип) × довжина\_масиву.*

### Приклад 2.

Елементом масиву `mas` присвоюються значення від 0 до 55.

```
#include<stdio.h>
int main()
{
 int mas[55];
 for(int i=0; i<55; i++)
 mas[i] = i;
 for(i=0; i<55; i++)
 printf("%d ", mas[i]);
 return 0;
}
```

### Приклад 3.

Програма працює з трьома масивами, перший з яких ініціалізується цілими числами, другий заповнюється випадковими числами, а елементи третього - сумами відповідних елементів двох масивів.

```
#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<conio.h>
int main ()
{
 srand(time(NULL));
 const int n=15;
 int arr1[] = {1,3,10,2,4,8,43,9,14,0,1,2,3,5,9};
 for (int i=0; i<n; i++)
 printf("%5d",arr1[i]);
 printf("\n-----\n");
```

```

int arr2[n];
for (int i=0; i<n; i++)
{
 arr2[i]=rand()%25+1;
 printf("%5d",arr2[i]);
}
printf("\n-----\n");

int arr3[n];
for (int i=0; i<n; i++)
{
 arr3[i]=arr1[i] + arr2[i];
 printf("%5d",arr3[i]);
}
printf("\n-----\n");
_getch();
return 0;
}

```

### Контрольні питання:

1. Яким чином можна згенерувати випадкове число?
2. Для чого існує функція rand()?
3. Яким чином генеруються цілі випадкові числа на певному інтервалі?
4. Як згенерувати дійсні випадкові числа на певному інтервалі?
5. Що таке масиви?
6. Як розташовуються елементи масивів у пам'яті?
7. Які бувають масиви за розмірністю та способом виділення їм пам'яті?
8. Як оголошують статичні масиви?
9. Як звернутись до першого та останнього елементу масиву?
10. Яким чином обчислюється кількість байтів пам'яті, виділених масиву?
11. Наведіть фрагмент програми (за допомогою операторів for, while та do-while), що обчислює суму, добуток, кількість парних елементів одновимірного масиву, найбільше та найменше значення?

## Тема № 12.

### Дослідження роботи одномірних масивів.

**Мета:** набуття навичок роботи з одновимірними масивами.

#### **Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

**Завдання.** Побудувати блок-схему, написати та налагодити програму. Написати тест до програми:

|    |                                                                                                                                                                                                                                                                          |
|----|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 1. | Дано масив А розміру N. Сформувати новий масив В того ж розміру, елементи якого визначаються в такий спосіб:<br>$V[i] = A[i]$ , якщо $A[i] < 5$ ,<br>$V[i] = A[i]/2$ інакше.                                                                                             |
| 2. | Дано два масиви А і В однакового розміру N. Сформувати новий масив С того ж розміру, кожен елемент якого дорівнює максимальному з елементів масивів А і В з тим же індексом, всі інші елементи дорівнюють 0.                                                             |
| 3. | Дано цілочисельний масив А розміру N. Переписати в новий цілочисельний масив В всі парні числа з вихідного масиву (у тім же порядку) і вивести розмір отриманого масиву В і його вміст.                                                                                  |
| 4. | Дано цілочисельний масив А розміру N ( $N \leq 15$ ). Переписати в новий цілочисельний масив В всі елементи з непарними порядковими номерами (1, 3, ...) і вивести розмір отриманого масиву В і його вміст. Умовний оператор не використовувати.                         |
| 5. | Дано цілочисельний масив А розміру N ( $N \leq 15$ ). Переписати в новий цілочисельний масив В всі елементи з порядковими номерами, кратними трьом (3, 6, ...), і вивести розмір отриманого масиву В і його вміст. Умовний оператор не використовувати.                  |
| 6. | Дано цілочисельний масив А розміру N. Переписати в новий цілочисельний масив В того ж розміру спочатку всі елементи вихідного масиву з парними номерами, а потім – з непарними:<br>$A[2], A[4], A[6], \dots, A[1], A[3], A[5], \dots$ . Умовний оператор не використати. |
| 7. | Дано масив А розміру N. Сформувати новий масив В того ж розміру за наступним правилом: елемент $V[i]$ дорівнює сумі елементів масиву А з номерами від 1 до i.                                                                                                            |
| 8. | Дано масив А розміру N. Сформувати новий масив В того ж розміру за наступним правилом: елемент $V[i]$ дорівнює середньому арифметичному елементів масиву А з номерами від 1 до i.                                                                                        |

|     |                                                                                                                                                                                                                                                                                                                        |
|-----|------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| 9.  | Дано масив А розміру N. Сформувати новий масив В того ж розміру за наступним правилом: елемент В[i] дорівнює сумі елементів масиву А з номерами від і до N.                                                                                                                                                            |
| 10. | Дано масив А розміру N. Сформувати новий масив В того ж розміру за наступним правилом: елемент В[i] дорівнює середньому арифметичному елементів масиву А з номерами від і до N.                                                                                                                                        |
| 11. | Дано масив А розміру N. Сформувати два нових масиви В й С: у масив В записати всі позитивні елементи масиву А, у масив С – всі від'ємні(дотримуючись вихідного порядку проходження елементів). Вивести спочатку розмір і вміст масиву В, а потім - розмір і зміст масиву С.                                            |
| 12. | Дано два масиви А й В розміру 5, елементи яких упорядковані за зростанням. Об'єднати ці масиви так, щоб результуючий масив С (розміру 10) залишився впорядкованим за зростанням.                                                                                                                                       |
| 13. | Дано три цілочисельних масиви А, В й С розміру N <sub>A</sub> , N <sub>B</sub> , N <sub>C</sub> відповідно, елементи яких впорядковані за спаданням. Об'єднати ці масиви так, щоб результуючий цілочисельний масив D (розміру N <sub>A</sub> + N <sub>B</sub> + N <sub>C</sub> ) залишився впорядкованим за спаданням. |
| 14. | Дано три цілочисельних масиви А, В й С розміру N <sub>A</sub> , N <sub>B</sub> , N <sub>C</sub> відповідно, елементи яких впорядковані за спаданням. Об'єднати ці масиви так, щоб результуючий цілочисельний масив D (розміру N <sub>A</sub> + N <sub>B</sub> + N <sub>C</sub> ) став впорядкованим за зростанням.     |
| 15. | Дано два масиви А й В розміру 5, елементи яких упорядковані за зростанням. Об'єднати ці масиви так, щоб результуючий масив С (розміру 10) став впорядкованим за спаданням.                                                                                                                                             |

### Методичні рекомендації

Часто виникає ситуація, коли кількість елементів масиву може варіюватися на етапі компіляції. При цьому перерахунок кількості елементів вручну призводить до втрати часу. С дозволяє уникнути даної проблеми.

Для визначення розміру масиву використовується наступна конструкція:

$$N = \text{sizeof}(\text{array}) / \text{sizeof}(\text{array}[0])$$

Де array – ім'я масиву, N - отриманий розмір масиву, sizeof (array) повертає розмір всього масиву в байтах, а sizeof (array [0]) – розмір першого елемента в байтах. Таким чином, поділивши розмір всього масиву на розмір одного елемента, ми отримаємо кількість елементів, тобто розмір масиву, який шукаємо.

Розглянемо наступний приклад: знайти максимальний елемент в масиві.

```
#include<stdio.h>
#include<conio.h>
#include<locale.h>
void main ()
{
 setlocale(LC_ALL, "Russian");
```

```
int b[] = {1, -5, 8, 7, 4, -7, 3, 0, -8, 6};
int n = sizeof(b)/sizeof(b[0]);
for (int i = 0; i < n; i++)
 printf("%d ", b[i]);
printf("\nn=%d", n);
getch();
}
```

**Контрольні питання:**

1. Що таке одновимірний масив?
2. Що таке елемент масиву та індекси?
3. Як описується одновимірний масив?
4. Які існують способи ініціалізації одновимірних масивів?

## Тема № 13.

### Сортування одновимірного масиву.

**Мета:** набуття практичних навиків роботи з одновимірними масивами, а саме сортування елементів масиву різними методами. Здійснення порівняння та аналізу ефективності використаних методів сортування.

#### **Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

#### **Завдання:**

- 1) Згенеруйте одновимірний масив і виведіть його на екран.
- 2) Відсортуйте масив кожним описаним методом.

### Методичні рекомендації

**Сортування масиву** – один з найбільш розповсюджених процесів обробки даних. Завдяки йому здійснюється розміщення об'єктів у визначеному порядку, наприклад, чисел за зростанням або за спаданням їх значень, прізвищ у алфавітному порядку тощо. Існують різні методи сортування, серед них – сортування обміном (метод "бульбашки", "шейкер-сортування"), сортування вибором, сортування вставками, швидке сортування, сортування Шелла, пірамідальне сортування, сортування обчисленням адреси, сортування порозрядним групуванням тощо. Ці методи відрізняються швидкістю отримання результату, складністю і універсальністю.

#### **Сортування обміном**

Метод "сортування обміном" ґрунтується на перестановці сусідніх елементів. Для впорядкування елементів масиву здійснюються повторні проходи по масиву. Переміщення елементів масиву здійснюється таким чином: масив переглядається зліва направо, здійснюється порівняння пари сусідніх елементів; якщо елементи в парі розміщені в порядку зростання, вони лишаються без змін, а якщо ні – міняються місцями. В результаті 1-го проходу найбільше число буде поставлено в кінець масиву. У 2-му проході операції виконуються над елементами з першого до (N-1)-ого, у 3-му – від першого до (N-2)-ого і т.п. Впорядкування масиву буде закінчено, якщо при проході масиву не виконається жодної перестановки елементів масиву. Факт перестановки фіксується за допомогою деякої змінної (fl), яка на початку має значення 0 і набуває значення 1, коли виконається перестановка в будь-якій парі.

| Масив до сортування | 20 | 5  | 67 | 3  | 4  | -2 | 5  | 10 |
|---------------------|----|----|----|----|----|----|----|----|
| 1                   | 5  | 20 | 3  | 4  | -2 | 5  | 10 | 67 |
| 2                   | 5  | 3  | 4  | -2 | 5  | 10 | 20 | 67 |
| 3                   | 3  | 4  | -2 | 5  | 5  | 10 | 20 | 67 |
| 4                   | 3  | -2 | 4  | 5  | 5  | 10 | 20 | 67 |
| 5                   | -2 | 3  | 4  | 5  | 5  | 10 | 20 | 67 |



```

const int n=8; int c, fl;
int array[]={20,5,67,3,4,-2,5,10};
do
{
 fl=0;
 for (int i=1;i<n;i++)
 if (array[i-1]>array[i])
 {
 c=array[i]; array[i]=array[i-1];
 array[i-1]=c;
 fl=1;
 }
} while (fl);

```

### Сортування методом вибору

Сутність методу: масив переглядається перший раз, знаходиться мінімальний елемент масиву і міняється місцями з першим елементом. Другий раз масив переглядається, починаючи з другого елементу. Знову знаходиться мінімальний елемент і міняється місцями з другим.

Даний процес виконується доти, поки не буде поставлений на місце N-1 елемент.

| Масив до сортування | 20 | 5 | 67 | 3 | 4  | -2 | 5  | 10 |
|---------------------|----|---|----|---|----|----|----|----|
| 1                   | -2 | 5 | 67 | 3 | 4  | 20 | 5  | 10 |
| 2                   |    | 3 | 67 | 5 | 4  | 20 | 5  | 10 |
| 3                   |    |   | 4  | 5 | 67 | 20 | 5  | 10 |
| 4                   |    |   |    | 5 | 5  | 20 | 67 | 10 |
| 5                   |    |   |    |   | 5  | 10 | 67 | 20 |
| 6                   |    |   |    |   |    | 10 | 20 | 67 |

```

const int n=8; int imin, a;
int array[]={20,5,67,3,4,-2,5,10};
for (int i=0;i<n-1;i++)
{
 imin=i;
 for (int j=i+1;j<n;j++)
 if (array[j]<array[imin]) imin=j;
 a=array[i];
 array[i]=array[imin];
 array[imin]=a;
 printf("\n");
}

```

### Сортування вставками

При використанні даного методу на i-му етапі відбувається "вставка" елемента  $a[i]$  в потрібну позицію серед елементів  $a[1], \dots, a[i-1]$ , які вже впорядковані. Після цієї вставки перші i елементів будуть впорядковані.

| Масив до сортування | 20 | 5  | 67 | 3  | 4  | -2 | 5  | 10 |
|---------------------|----|----|----|----|----|----|----|----|
| 1                   | 5  | 20 |    |    |    |    |    |    |
| 2                   | 5  | 20 | 67 |    |    |    |    |    |
| 3                   | 3  | 5  | 20 | 67 |    |    |    |    |
| 4                   | 3  | 4  | 5  | 20 | 67 |    |    |    |
| 5                   | -2 | 3  | 4  | 5  | 20 | 67 |    |    |
| 6                   | -2 | 3  | 4  | 5  | 5  | 20 | 67 | 10 |
| 7                   | -2 | 3  | 4  | 5  | 5  | 10 | 20 | 67 |

```

const int n=8; int c;
int array[]={20,5,67,3,4,-2,5,10};
for (int i=1;i<n;i++)
{
 c=array[i];
 for (int j=i-1;j>=0 && array[j]>c;j--)
 {
 array[j+1]=array[j];
 array[j]=c;
 }
}

```

### Сортування методом Шелла

Головний недолік простих методів – обмін ведеться в основному між сусідніми елементами. Тому бажано робити якомога ширші обміни.

Метод Шелла – метод сортування включеннями з відстанями, що зменшуються.

На першому етапі масив уявно ділиться на підмасиви наприклад  $size = 8$ ,  $step = size / 2 = 4$ , кожен з них впорядковується окремо.

На другому етапі підмасиви утворюються елементами через один.

Після цього весь масив сортується разом. За рахунок попередніх етапів він стає вже близьким до відсортованого, тому обмінів необхідно вже не так багато. Цей метод дає кращі результати, якщо розподіл на підмасиви роблять кроками, що не є степенями двійки, а, навпаки, не є множниками один одного.

Масив до сортування     **20   5   67   3   4   -2   5   10**

|   |    |    |   |   |    |    |    |    |
|---|----|----|---|---|----|----|----|----|
| 1 | 4  | -2 | 5 | 3 | 20 | 5  | 67 | 10 |
|   |    |    |   |   |    |    |    |    |
| 2 | 4  | -2 | 5 | 3 | 20 | 5  | 67 | 10 |
|   |    |    |   |   |    |    |    |    |
| 3 | -2 | 3  | 4 | 5 | 5  | 20 | 10 | 67 |
|   |    |    |   |   |    |    |    |    |

```

int step = size / 2;
while (step > 0)
{
 for (int i = 0; i < (size - step); i++)
 {
 int j = i;
 while (j >= 0 && a[j] > a[j + step])
 {
 int c = a[j];
 a[j] = a[j + step];
 a[j + step] = c;
 j--;
 }
 }
 step = step / 2;
}

```

Функція `clock()` для виміру часу виконання програми.

```

#include <time.h>
float dd;
clock_t start, finish;
start = cLock(); // час початку виконання програми

// ... код програми...

finish = cLock(); // час завершення виконання програми
dd = (float)(finish - start) / CLOCKS_PER_SEC;
// час виконання програми

```

### Контрольні питання:

1. Які методи сортування ви знаєте? В чому їх сутність?
2. Поясніть алгоритм методу сортування *обміном*.
3. Наведіть приклад фрагмента коду, що здійснює сортування елементів масиву обміном у порядку зростання (починати сортування з першого елемента)
4. Поясніть алгоритм методу сортування *вставками*.
5. Наведіть приклад фрагмента коду, що здійснює сортування цілих чисел вставками у порядку зменшення (починати сортування з першого елемента)
6. Поясніть алгоритм методу сортування методом *вибору*.
7. Наведіть схему роботи сортування вибором (сортування починати з початку).
8. Наведіть приклад фрагмента коду, що здійснює сортування дійсних чисел вибором у порядку зменшення (починати сортування з кінця)
9. Наведіть схему роботи сортування методом обміну.
10. Як оцінюються методи сортування?
11. Наведіть алгоритм сортування методом Шелла. Які його переваги?

## Тема № 14. Опрацювання двовимірних масивів.

**Мета:** набуття практичних навичок роботи з двовимірними масивами.

### **Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

### Зміст роботи

**Завдання.** Написати програму згідно варіанту. Результати вивести на екран у зручному для сприйняття вигляді.

|          |                                                                                                                                                                                         |
|----------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>1</b> | Сформуйте двовимірний масив, який складається з 5 рядків і 3 стовпців за правилом $a[i,j]=i-j*j$ , і виведіть його на екран.<br>Обчисліть суму квадратів елементів другого рядка.       |
| <b>2</b> | Сформуйте двовимірний масив, який складається з 4 рядків і 3 стовпців за правилом $b[i,j]=i-j$ , і виведіть його на екран<br>Знайдіть кількість ненульових елементів у масиві.          |
| <b>3</b> | Сформуйте двовимірний масив, який складається з 3 рядків і 4 стовпців за правилом $c[i,j]=j+i$ та виведіть його на екран.<br>Обчисліть середнє арифметичне елементів таблиці.           |
| <b>4</b> | Сформуйте двовимірний масив, який складається з 4 рядків і 4 стовпців за правилом $m[i,j]=i*j-3$ та виведіть його на екран.<br>Обчисліть добуток елементів третього стовпчика.          |
| <b>5</b> | Сформуйте двовимірний масив, який складається з 3 рядків і 6 стовпців за правилом $a[i,j]=2i-j$ та виведіть його на екран.<br>Знайти добуток усіх елементів таблиці, що більші від 1.   |
| <b>6</b> | Сформуйте двовимірний масив, який складається з 4 рядків і 4 стовпців за правилом $a[i,j]=i-2j$ та виведіть його на екран.<br>Замініть ненульові елементи на 1.                         |
| <b>7</b> | Сформуйте двовимірний масив, який складається з 5 рядків і 3 стовпців за правилом $m[i,j]=i-j*i$ та виведіть його на екран.<br>Знайти суму елементів першого рядка.                     |
| <b>8</b> | Сформуйте двовимірний масив, який складається з 4 рядків і 5 стовпців за правилом $a[i,j]=2i-(i+j)$ та виведіть його на екран.<br>Знайдіть добуток елементів п'ятого стовпчика таблиці. |
| <b>9</b> | Сформуйте двовимірний масив, який складається з 5 рядків і 4 стовпців за правилом $c[i,j]=2-i-j$ та виведіть її на екран.<br>Знайти добуток елементів п'ятого рядка таблиці.            |

|           |                                                                                                                                                                                       |
|-----------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>10</b> | Сформууйте двовимірний масив, який складається з 3 рядків і 5 стовпців за правилом $c[i,j]=i*i-j*3$ та виведіть його на екран.<br>Знайти середнє арифметичне елементів другого рядка. |
| <b>11</b> | Сформууйте двовимірний масив, який складається з 3 рядків і 3 стовпців за правилом $a[i,j]=i*i-j*j$ та виведіть його на екран.<br>Знайдіть кількість від'ємних елементів таблиці.     |
| <b>12</b> | Сформууйте двовимірний масив, який складається з 5 рядків і 3 стовпців за правилом $a[i,j]=i-j*j$ та виведіть його на екран.<br>Знайдіть кількість додатних елементів таблиці.        |
| <b>13</b> | Сформууйте двовимірний масив, який складається з 3 рядків і 4 стовпців за правилом $a[i,j]=2i-3j$ та виведіть його на екран.<br>Знайдіть суму елементів четвертого стовпчика.         |
| <b>14</b> | Сформууйте двовимірний масив, який складається з 4 рядків і 3 стовпців за правилом $c[i,j]=4i-2j$ та виведіть його на екран.<br>Знайдіть суму елементів четвертого рядка.             |
| <b>15</b> | Сформууйте двовимірний масив, який складається з 5 рядків і 5 стовпців за правилом $m[i,j]=2ij-i$ та виведіть його на екран.<br>Знайти добуток елементів п'ятого рядка таблиці.       |

### Методичні рекомендації

Крім одновимірних масивів, таких, де позиція елемента визначається за допомогою одного індексу, у практиці розв'язання задач часто застосовуються багатовимірні масиви. У них позиція елемента визначається записом декількох індексів. Найбільш розповсюджені *двовимірні масиви* або *матриці*. Матриці являють собою порядковий запис декількох одновимірних масивів. Місце розташування кожного елемента визначається за допомогою двох індексів - номера рядка і номера стовпця, тобто порядкового номера в рядку. Індеси двовимірних масивів записуються в квадратних дужках і нумерація індексів починається з нуля (**0**).

Наприклад, двовимірний масив цілих чисел `int a[3][4]`, що має три рядки та чотири стовпці. Вигляд двовимірного масиву (матриці) `int a[3][4]`:

|                      |                      |                      |                      |
|----------------------|----------------------|----------------------|----------------------|
| <code>a[0][0]</code> | <code>a[0][1]</code> | <code>a[0][2]</code> | <code>a[0][3]</code> |
| <code>a[1][0]</code> | <code>a[1][1]</code> | <code>a[1][2]</code> | <code>a[1][3]</code> |
| <code>a[2][0]</code> | <code>a[2][1]</code> | <code>a[2][2]</code> | <code>a[2][3]</code> |

У пам'яті комп'ютера масив розташовується безперервно за рядками:

`a [0][0], a [0][1], a [0][2], a [0][3], a [1][0], a [1][1], a [1][2], a [1][3], ...a [2][3]`.

Двовимірні (і багатовимірні) масиви оголошуються наступним чином:

`int mas [2][5] = {1, 5, 3, 7, 4, 10, 11, 13, 14, 25 };`

```
int mas [][5] = {1, 5, 3, 7, 4, 10, 11, 13, 14, 25 };
```

`int mas [ ][5] = {{ 1, 5, 3, 7, 4 }, {10, 11, 13, 14, 25}}`; тобто масив задається або списком елементів у тому порядку, и якому вони розташовані у пам'яті, або подається як масив масивів, кожний з яких поміщається в свої фігурні дужки "{}". При оголошенні і одночасному ініціюванні багатовимірних масивів можна опускати кількість індексів тільки першого виміру. Якщо ініціювання не здійснюється під час оголошення масиву, то кількість індексів треба вказувати явно.

Загальна форма оголошення двовимірного масиву:

```
<тип елементів><назва масиву>[<розмірність1>][<розмірність2>;
```

Для здійснення введення-виведення, а також для обробки елементів двовимірного масиву у програмі слід передбачати організацію двох циклів: один – для задавання значень індексу рядків, другий – індексу стовпців.

### Контрольні питання:

1. Що таке двовимірний масив?
2. Які дані можуть бути записані у масив?
3. Які оператори використовуються для опису двовимірних масивів?
4. Як визначити індекс двовимірного масиву?
5. Яким чином здійснюється доступ до елементів двовимірного масиву?
6. Наведіть приклади з життя, в яких можна було б використовувати двовимірні масиви.

**Тема № 15.**  
**Робота з двовимірними масивами.**

**Мета:** набуття практичних навичок роботи з двовимірними масивами.

**Література:**

C/C++: Теорія та практика. Навчально-методичний посібник. / В. В. Войтенко, А. В. Морозов. – Житомир: ЖДТУ, 2004. – 324 с.

**Зміст роботи**

**Завдання.** Дана квадратна матриця.

|     |                                                                     |
|-----|---------------------------------------------------------------------|
| 1.  | Розмістити елементи парних рядків у порядку зростання.              |
| 2.  | Розмістити елементи непарних рядків у порядку зростання.            |
| 3.  | Розмістити елементи парних рядків у порядку зростання.              |
| 4.  | Розмістити елементи непарних стовпців у порядку зростання.          |
| 5.  | Розмістити елементи парних рядків у порядку спадання.               |
| 6.  | Розмістити елементи непарних рядків у порядку спадання.             |
| 7.  | Розмістити елементи парних стовпців у порядку спадання.             |
| 8.  | Розмістити елементи непарних стовпців у порядку спадання.           |
| 9.  | Розмістити елементи головної діагоналі матриці у порядку зростання. |
| 10. | Розмістити елементи головної діагоналі матриці у порядку спадання.  |
| 11. | Розмістити елементи бічної діагоналі матриці у порядку зростання.   |
| 12. | Розмістити елементи бічної діагоналі матриці у порядку спадання.    |
| 13. | Розмістити елементи вказаних рядків у порядку зростання.            |
| 14. | Розмістити елементи вказаних рядків у порядку спадання.             |
| 15. | Розмістити елементи вказаних стовпців у порядку спадання.           |

**Контрольні запитання:**

1. Що таке масив?
2. Що таке розмірність масиву? Чи існують обмеження на розмірність масиву?
3. Якого типу можуть бути елементи масиву?
4. Якого типу можуть бути індекси елементів масиву?
5. Якими способами може бути заповнений масив? Наведіть приклади.
6. Як визначити мінімальний обсяг пам'яті, що відводиться під масив?
7. Які дії виконують зазвичай над елементами масиву?
8. Чи може масив бути елементом масиву?
9. Нехай елементами масиву А (а [1], а [2], а [3], а [4]) є відповідно  $x$ ,  $-x$ ,  $x^2$ ,  $-x^2$ . Чому дорівнюватиме значення виразу  $a[-a[a[3]-2]] + a[-a[a[3]]]$  при  $x = 2$ ?
10. Чи можна виконувати обхід двовимірного масиву, організувавши зовнішній цикл по одному стовпчику, а внутрішній – по рядках?

Навчально-методичне видання

**МЕТОДИЧНІ РЕКОМЕНДАЦІЇ  
ДЛЯ ВИКОНАННЯ ЛАБОРАТОРНИХ РОБІТ  
З ДИСЦИПЛІНИ**

**“ОСНОВИ ПРОГРАМУВАННЯ”**

**ЧАСТИНА 1**

Підготували  
**Марчук Галина Вікторівна**  
**Левківський Віталій Леонідович**  
**Чижмотря Олексій Володимирович**

Комп’ютерний набір та верстка: Марчук Г.В., Левківський В.Л., Чижмотря О.В.

Підписано до друку \_\_.\_\_.19. Формат 60×84/16.

Ум. друк. арк. \_\_\_\_\_. Зам. \_\_ офс.

Безкоштовно

Друкарня ЖДГУ