

**Практична робота №1**

**Тема:** програмування задач лінійної структури.

**Мета:** навчитись складати алгоритми задач лінійної структури та навчитись програмувати задачі з послідовним виконанням програми.

**Завдання №1**

**Варіант №10**

Дано два цілі числа. Знайти їx серсдне арифметичне.

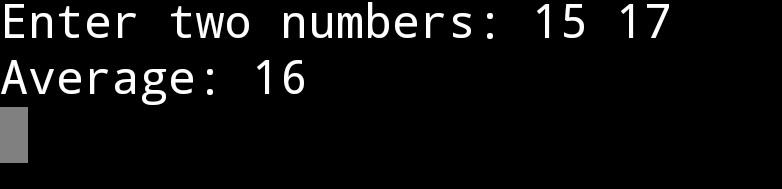
int main() {

int n1, n2;

cout << "Enter two numbers: ";

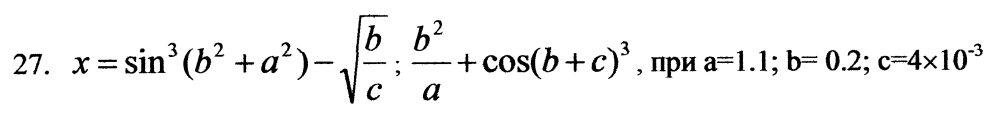
cin >> n1 >> n2;

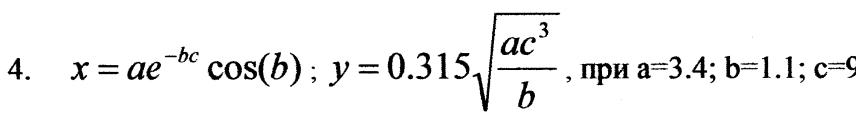
}



**Завдання №2**

**Варіант №27, 4**

Обчислити 



int main() {

double a = 1.1, b = 0.2, c = 0.004;

double x = (pow(sin(pow(b, 2) + pow(a, 2)), 3)) - (sqrt(b / c));

double y = (pow(b, 2) / a) + (cos(pow((b + c), 3)));

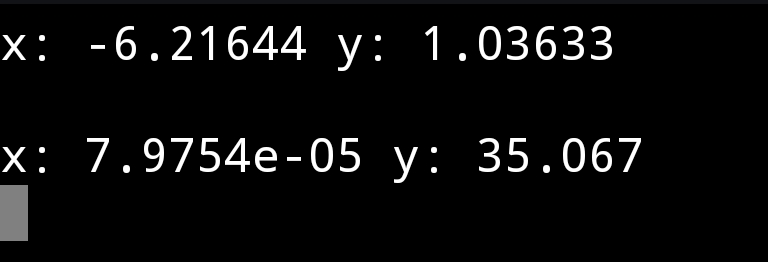
const double e = 2.71;

double a1 = 3.4, b1 = 1.1, c1 = 9;

double x1 = a1 \* pow(e, -(b1 \* c1)) \* cos(b1);

double y1 = 0.315 \* sqrt((a1 \* pow(c1, 3)) / b);

}



**Завдання №3**

**Варіант №11**

1. Дана відстань в метрах. Знайти число повних кілометрів в нім.

int main() {

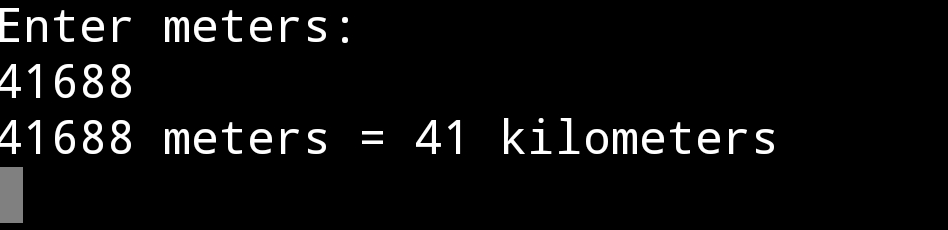
int meters, kilometers;

cout << "Enter meters: " << endl;

cin >> meters;

kilometers = meters / 1000;

}



2. Дано чотиризначне число. Знайти добуток його цифр.

int main() {

int num;



cout << "Enter 4 digit number: " << endl;

cin >> num;

if (num < 1000 || num > 9999) {

cout << "ERROR" << endl;

return 1;

}

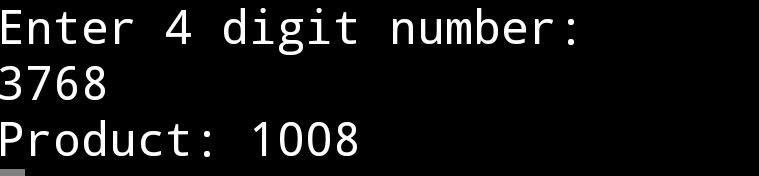
int digit1 = num / 1000;

int digit2 = (num / 100) % 10;

int digit3 = (num / 10) % 10;

int digit4 = num % 10;

}



3. Задано цілі числа h, m (0< h< =12, 0< m <= 59), які вказують момент часу: "h годин, m, хвилин” Визначити найменший час (число повних хвилин), який повинен пройти до Того моменту, коли годинникова і хвилинна стрілки на циферблаті розташуються перпендикулярно один одному.

int main() {

int h, m;

cout << "Enter hours and minutes: ";

cin >> h >> m;

for (int t = 0; t < 720; ++t) {

int new\_min = (m + t) % 60;

int new\_hour = (h + (m + t) / 60) % 12;

if (new\_hour == 0) new\_hour = 12;

double minute\_angle = new\_min \* 6.0;

double hour\_angle = new\_hour \* 30.0 + new\_min \* 0.5;

double diff = fabs(minute\_angle - hour\_angle);

if (fabs(diff - 90.0) < 1e-6 || fabs(diff - 270.0) < 1e-6) {

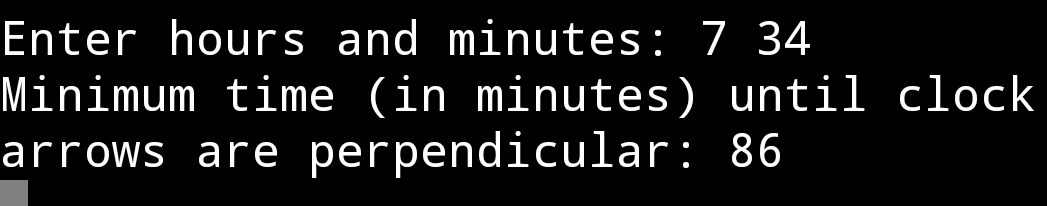
break;

}

}

return 0;

}



**Практична робота №2**

**Тема:** розробка програм з розгалуженням.

**Мета:** набути практичних навичок введення−виведення даних, навичок у використанні операторів керування; створення і налагодження програм з розгалуженням.

**Завдання №1**

1. Введіть два числа. Виведіть повідомлення, яке з чисел більше.

int main() {

int i, j;

cout << "Enter two numbers: ";



cin >> i >> j;

if (i > j) {

cout << "First number greater than second\n";

}

else if (i < j) {

cout << "Second number greater than first\n";

}

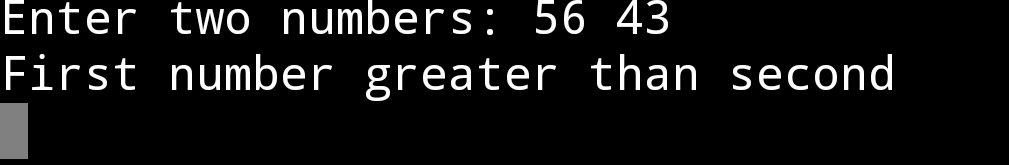
else {

cout << "Numbers are equal\n";

}

return 0;

}



2. Обчислити вартість покупки з урахуванням знижки. Знижка 3% надається, якщо сума покупки перевищує 500 грн., 5% - якщо сума більша, ніж 1000 грн.

int main() {

while (true) {

float stPrice, disPrice;

cout << "Enter price: ";

cin >> stPrice;

if (stPrice < 500) {

disPrice = stPrice;

} else if (stPrice >= 500 && stPrice < 1000) {

disPrice = stPrice \* 0.97;

} else if (stPrice >= 1000) {

disPrice = stPrice \* 0.95;

} else {

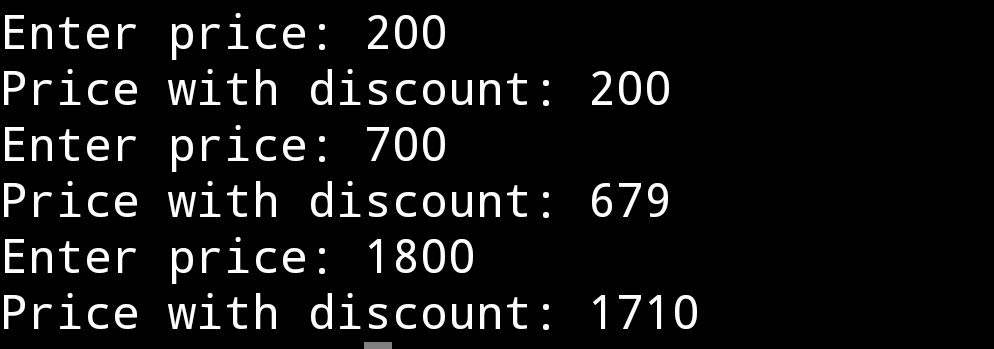
cout << "Error\n";

}

}

return 0;

}



3. Потрібно написати програму, що переводять температуру за шкалою Цельсія TC (°С) в температуру за Кельвіном TK (К). Значення TC вводиться користувачем з клавіатури.

int main() {

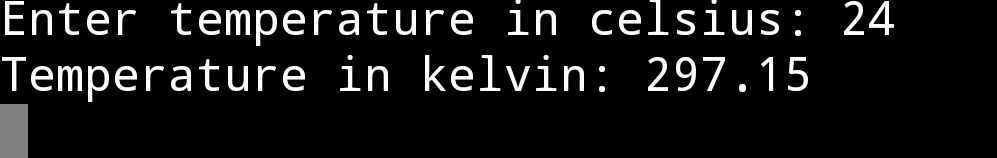
float celsiusTemp;

cout << "Enter temperature in celsius: ";

cin >> celsiusTemp;

return 0;

}



4. Користувач вводить порядковий номер пальця руки. Необхідно показати його

назву на екран або повідомлення про помилку, якщо введені невірні дані.

int main() {

while (true) {

int finger;

cout << "Enter finger number: ";

cin >> finger;

switch (finger) {



case 1:

cout << "Little finger\n";

break;

case 2:

cout << "Ring finger\n";

break;

case 3:

cout << "Middle finger\n";

break;

case 4:

cout << "Index finger\n";

break;

case 5:

cout << "Big finger\n";

break;

default:

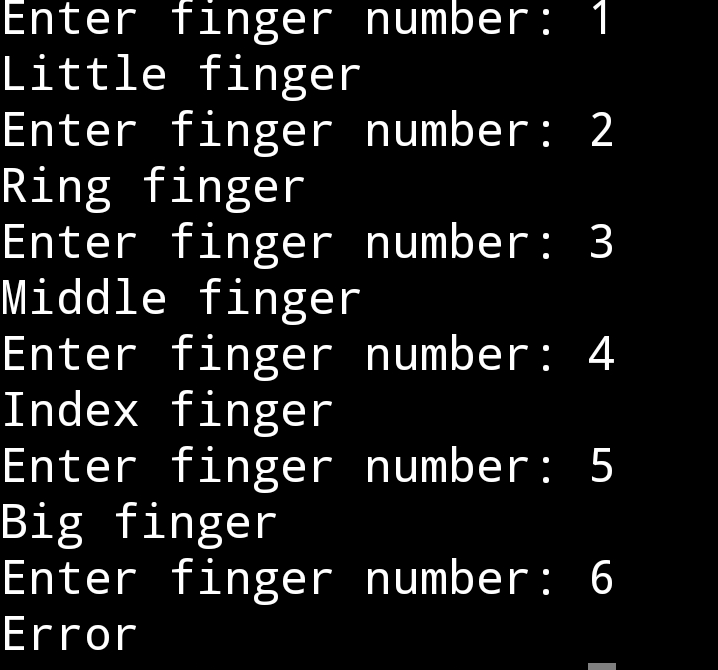
cout << "Error\n";

}

}

return 0;

}



**Завдання №2**

1. Обчислити частку двох чисел. Програма повинна перевіряти правильність уведених даних і, якщо вони неправильні (дільник дорівнює нулю), видавати повідомлення про помилку.

int main() {

double divident, divisor, quotient;

cout << "Enter dividend and divisor: ";

cin >> divident >> divisor;

if (divisor != 0) {

quotient = divident / divisor;

}

else {

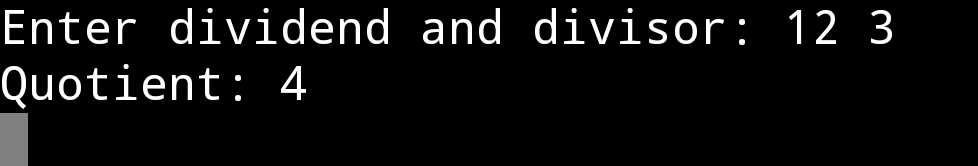
quotient = 0;

cout << "Error\n";

}

return 0;

}



2. Обчислити вартість покупки з урахуванням знижки. Знижка 10% надається, якщо сума покупки перевищує 1000 грн.

int main() {

float price2, disprice2;

cout << "Enter price: ";

cin >> price2;

if (price2 > 1000) {

disprice2 = price2 \* 0.9;



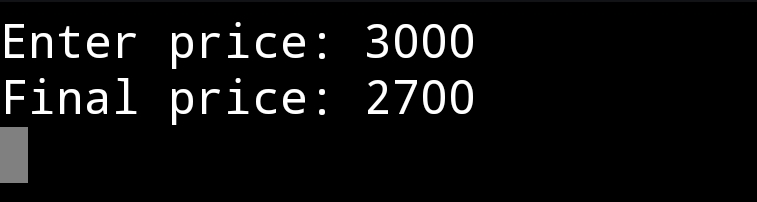
} else {

disprice2 = price2;

}

return 0;

}



3. Користувач вводить з клавіатури три цілих числа a, b, c. Необхідно вивести на екран найбільше з цих чисел.

int main() {

int a, b, c, largestT3;

cout << "Enter 3 integers: ";

cin >> a >> b >> c;

if (a >= b) {

if (a >= c) {

largestT3 = a;

} else {

largestT3 = c;

}

} else {

if (b >= c) {

largestT3 = b;

} else {

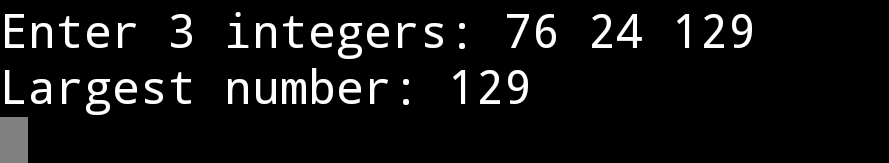
largestT3 = c;

}

}

return 0;

}



4. Напишіть програму, яка запрошує у користувача номер дня тижня, потім виводить назву дня тижня або повідомлення про помилку, якщо введені невірні дані.

int main() {

int dayNum;

cout << "Enter day number: ";

cin >> dayNum;

switch (dayNum) {

case 1:

cout << "Monday" << endl;

break;

case 2:

cout << "Tuesday" << endl;

break;

case 3:

cout << "Wednesday" << endl;

break;

case 4:

cout << "Thursday" << endl;

break;

case 5:

cout << "Friday" << endl;

break;

case 6:

cout << "Saturday" << endl;

break;

case 7:

cout << "Sunday" << endl;



break;

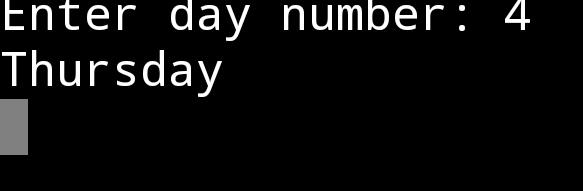
default:

cout << "Invalid day number" << endl;

}

return 0;

}



5. Введіть два числа. Виведіть повідомлення, яке з чисел більше.

int main() {

int i, j;

cout << "Enter two numbers: ";

cin >> i >> j;

if (i > j) {

}

else if (i < j) {

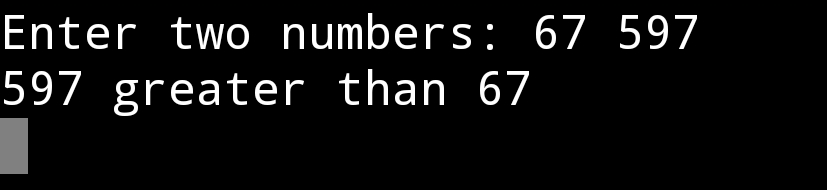
}

else {

}

return 0;

}



**Практична робота №3**

**Тема:** використання оператора-перемикача switch.

**Мета:** набути практичних навичок введення-виведення даних, навичок увикористанні операторів керування; створення і налагодження програм з розгалуженням.

**Завдання №1**

1. Напишіть програму, яка запитує у користувача номер дня тижня, а потім виводить назву дня тижня. При введенні неправильних даних програма повинна видати відповідне повідомлення.

int main() {

int dayNum;

cout << "Enter day number: ";

cin >> dayNum;

switch (dayNum) {

case 1:

cout << "Monday" << endl;

break;

case 2:

cout << "Tuesday" << endl;

break;

case 3:

cout << "Wednesday" << endl;



break;

case 4:

cout << "Thursday" << endl;

break;

case 5:

cout << "Friday" << endl;

break;

case 6:

cout << "Saturday" << endl;

break;

case 7:

cout << "Sunday" << endl;

break;

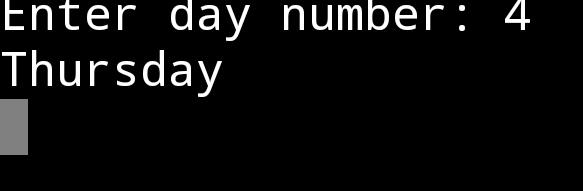
default:

cout << "Invalid day number" << endl;

}

return 0;

}



**2. Написати програму, що вимагає введення часу дня і, залежно від уведеного значення, бажає доброго ранку, доброго дня, доброго вечора або спокійної ночі.**

**int main() {**

**int hour;**

**cout << "Enter hour of the day (0-23): ";**

**cin >> hour;**

**if (hour >= 5 && hour < 12)**

**cout << "Good morning!" << endl;**

**else if (hour >= 12 && hour < 17)**

**cout << "Good afternoon!" << endl;**

**else if (hour >= 17 && hour < 21)**

**cout << "Good evening!" << endl;**

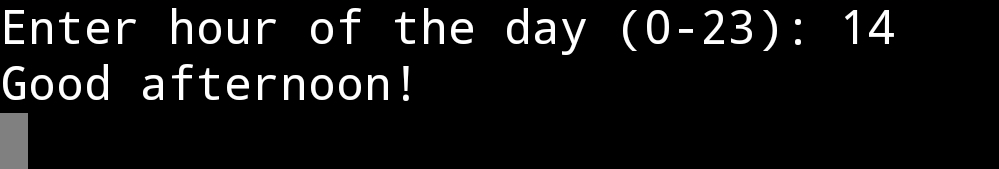
**cout << "Good night!" << endl;**

**else**

**cout << "Invalid time input." << endl;**

**return 0;**

**}**



**3.** Написати програму для обчислення вартості розмови по телефону з

урахуванням 20% знижки, що надається у суботу і неділю. Увести тариф,

тривалість розмови й номер дня тижня.

**int main() {**

**float tariff, duration;**

**string day;**

**cout << "Enter tariff per minute: ";**

**cin >> tariff;**

**cout << "Enter call duration (in minutes): ";**

**cin >> duration;**

**cout << "Enter day of the week (e.g., Monday, Saturday): ";**

**cin >> day;**

**float cost = tariff \* duration;**

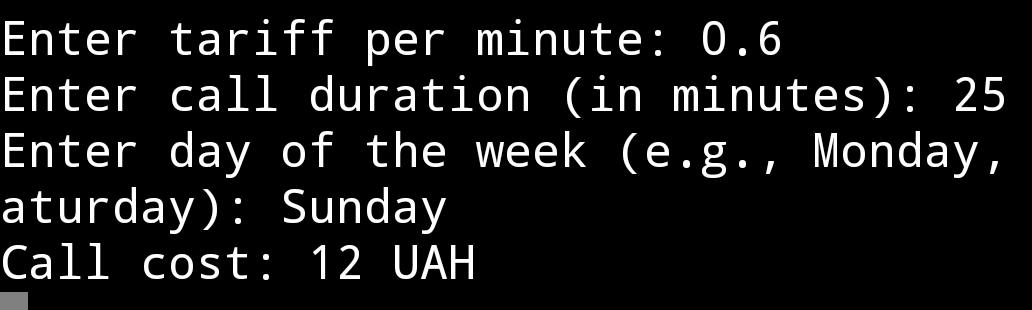
**if (day == "Saturday" || day == "Sunday")**

**cost \*= 0.8;**

**return 0;**



**}**



**Завдання №2**

1. Напишіть програму, яка запитує у користувача номер місяця і потім виводить відповідну назву пори року.

int main() {

int month;

cout << "Enter month number (1-12): ";

cin >> month;

if (month == 12 || month == 1 || month == 2)

cout << "Winter" << endl;

else if (month >= 3 && month <= 5)

cout << "Spring" << endl;

else if (month >= 6 && month <= 8)

cout << "Summer" << endl;

else if (month >= 9 && month <= 11)

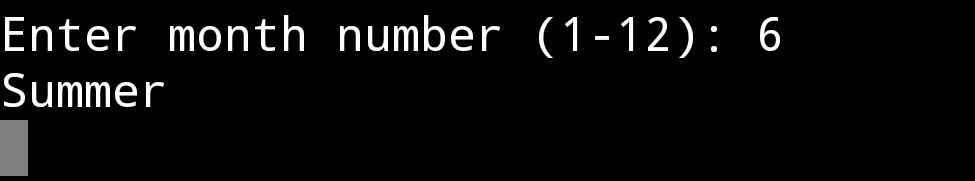
cout << "Autumn" << endl;

else

cout << "Data entry error" << endl;

**return 0;**

}



2. Обчислити вартість покупки з урахуванням знижки. Знижка 3% надається, якщо сума покупки перевищує 500 грн., 5% - якщо сума більша, ніж 1000 грн.

int main() {

while (true) {

float stPrice, disPrice;

cout << "Enter price: ";

cin >> stPrice;

if (stPrice < 500) {

disPrice = stPrice;

} else if (stPrice >= 500 && stPrice < 1000) {

disPrice = stPrice \* 0.97;

} else if (stPrice >= 1000) {

disPrice = stPrice \* 0.95;

} else {

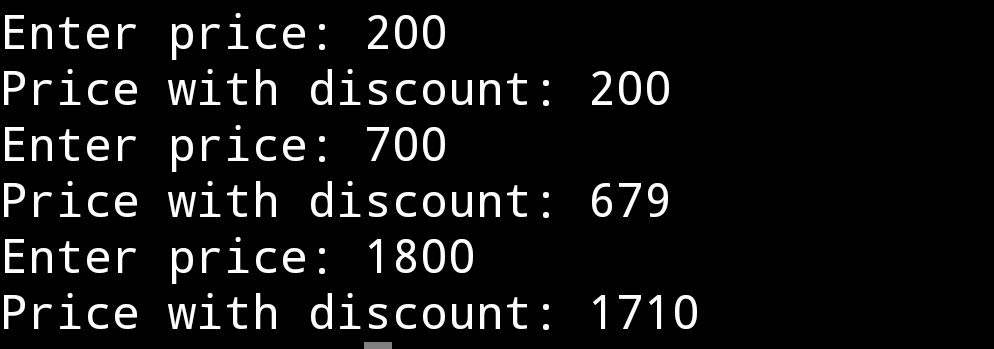
cout << "Error\n";

}

}

return 0;

}



**3. Створити програму калькулятор, в якій при виборі першого пункту обчислюється сума двох раніше введених цифр, при виборі другого пункту – обчислюється різниця, третього – добуток, четвертого частка.**

**int main() {**

**double a, b;**

**int choice;**



**cout << "Enter two numbers: ";**

**cin >> a >> b;**

**cout << "Choose operation:\n1. +\n2. -\n3. \*\n4. /\nEnter choice: ";**

**cin >> choice;**

**switch (choice) {**

**case 4:**

**if (b != 0)**

**else**

**cout << "Error: Division by zero" << endl;**

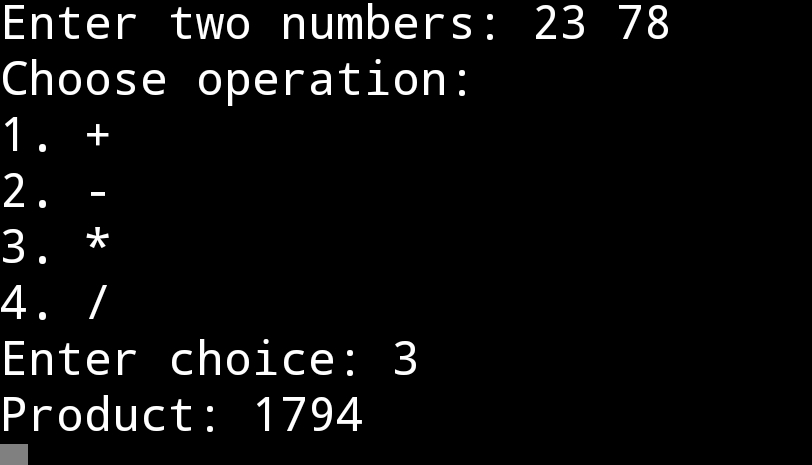
**break;**

**default: cout << "Invalid choice" << endl;**

**}**

**return 0;**

**}**



**Практична робота №4**

**Тема**: Програми з циклом із лічильником

**Мета**: Навчитись створювати програми з циклом і з лічильником

**Завдання №2**

**Варіант №1**

1. Написати програму, яка виводить на екран ваше ім'я 10 разів.

int main() {

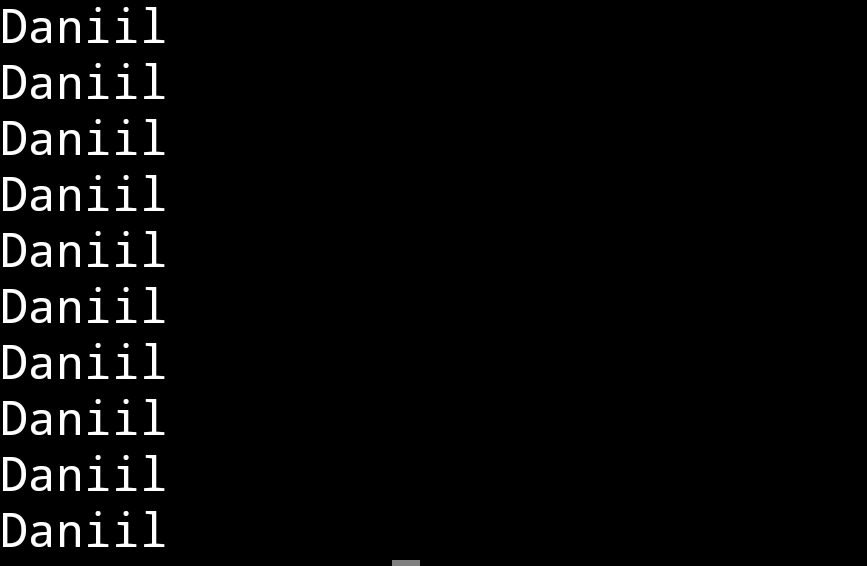
for (int n = 0; n < 10; n++) {

cout << "Daniil\n";

}

return 0;

}





2. Надрукувати всі натуральні числа від 1 до введеного з клавіатури n та їх квадрати у вигляді таблиці.

int main() {

int num;

cout << "Enter number: ";

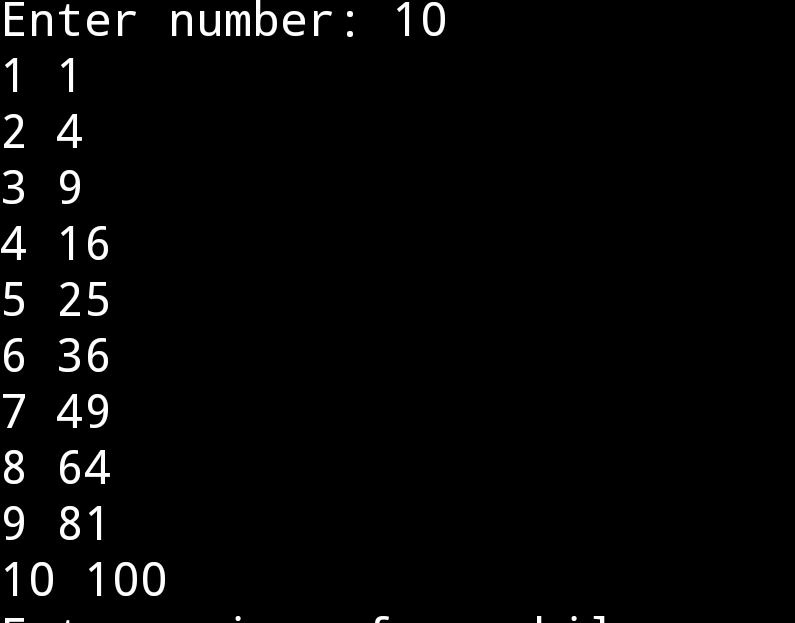
cin >> num;

for (int i = 1; i <= num; i++) {

}

return 0;

}



3. Написати програму, яка виводить на екран таблицю вартості, наприклад, яблук в діапазоні від 100 г до 1 кг з кроком 100 г.

int main() {

float pricekg;

cout << "Enter price of one kilogram: ";

cin >> pricekg;

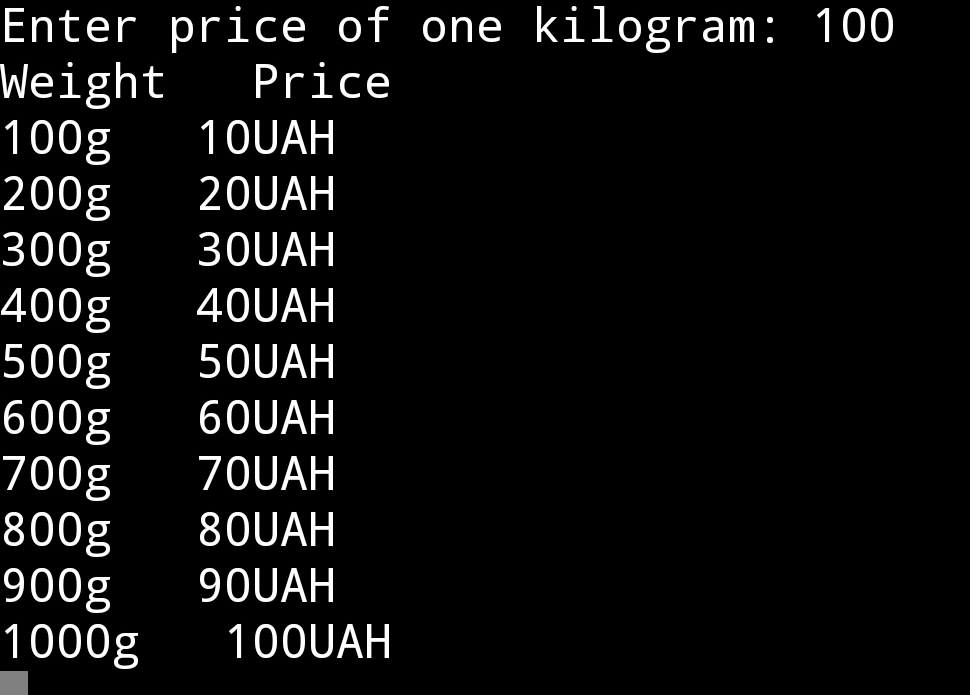
cout << "Weight Price\n";

for (int grl = 1; grl <= 10; grl++) {

}

return 0;

}



4. Напишіть програму, яка виводить на екран зображення шахової дошки.

int main() {

int size;

cout << "Enter size of board: ";

cin >> size;

for (int row = 0; row < size; ++row) {

for (int col = 0; col < size; ++col) {

if ((row + col) % 2 == 0) {

cout << " ";

} else {

cout << "#";

}

}

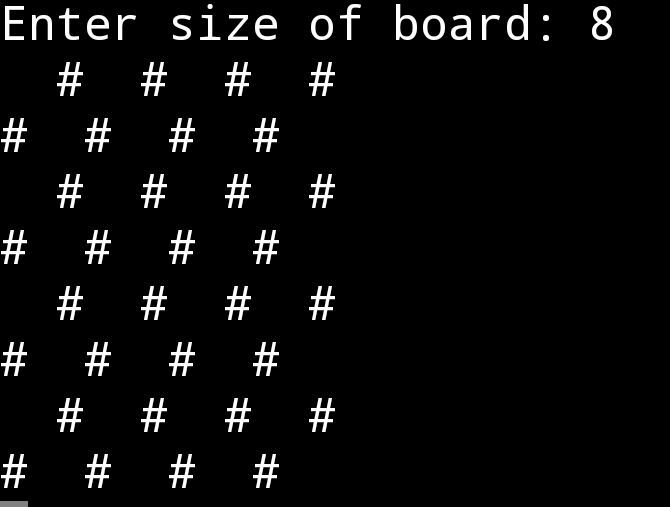
cout << endl;



}

return 0;

}



**Завдання №2**

**Варіант №2**

1. Написати програму, яка виводить на екран ваше прізвище 10 разів.

int main() {

for (int n = 0; n < 10; n++) {

cout << "Solovey\n";

}

return 0;

}



2. Капосний папуга навчився висмикувати у дідуся Івана волосся, яке ще залишилось у того на голові. Почавши з однієї волосини, він кожен день збільшував порцію вдвічі. Через скільки днів дідусеві не знадобиться гребінець, якщо на початку в нього було аж N волосин.

int main() {

int hair, day;

cout << "Enter number of hair: ";

cin >> hair;

for (int phs = 0; (hair - pow(2, phs)) > 0; phs++) {

int newHair;

if ((hair - pow(2, phs)) > 0) {

newHair = hair - pow(2, phs);

hair = hair - pow(2, phs);

} else if ((hair - pow(2, phs)) <= 0) {

newHair = 0;

hair = 0;

}

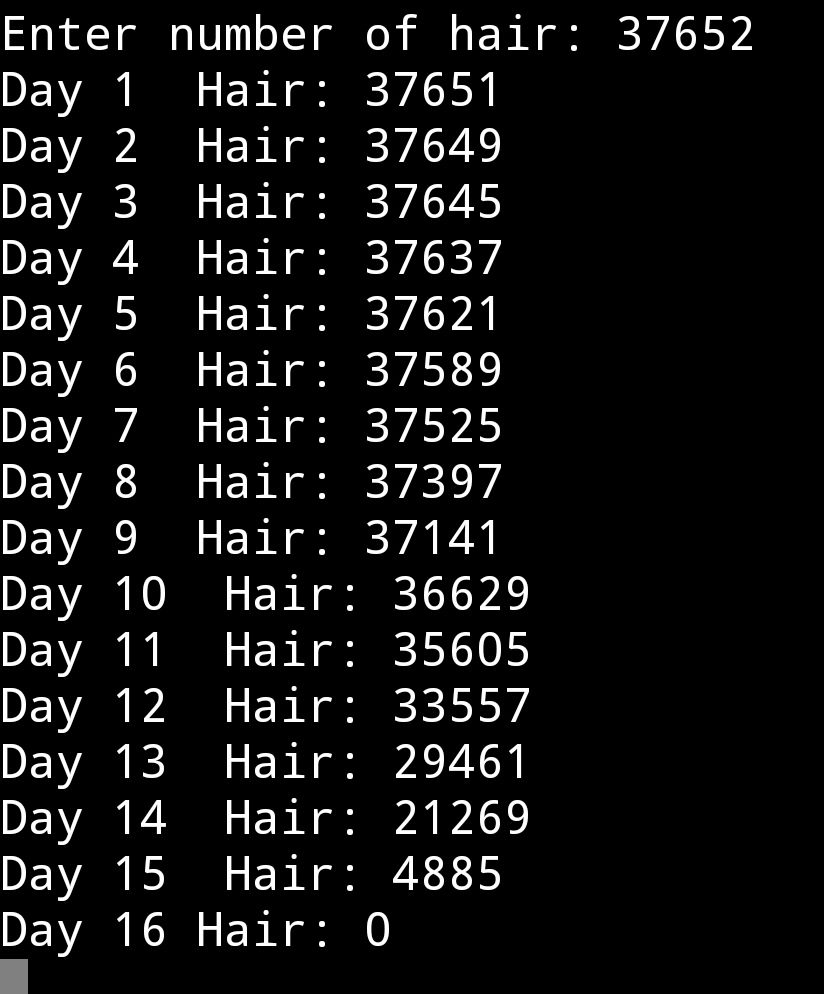
day = phs + 2;

}

return 0;

}





3. Написати програму, яка виводить таблицю степенів двійки від нульової до десятої.

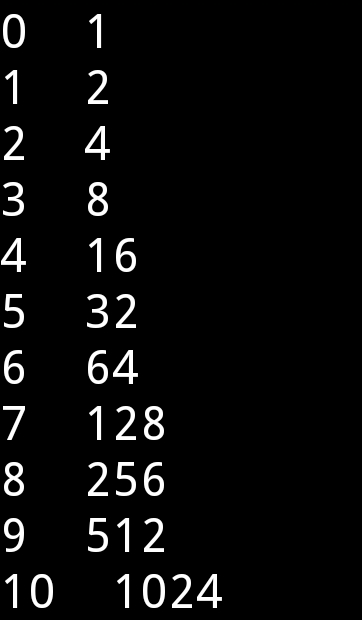
**int main() {**

**for (int p = 0; p <= 10; p++) {**

**}**

return 0;

}



4. Написати програму, яка виводить таблицю квадратів перших десяти цілих позитивних чисел.

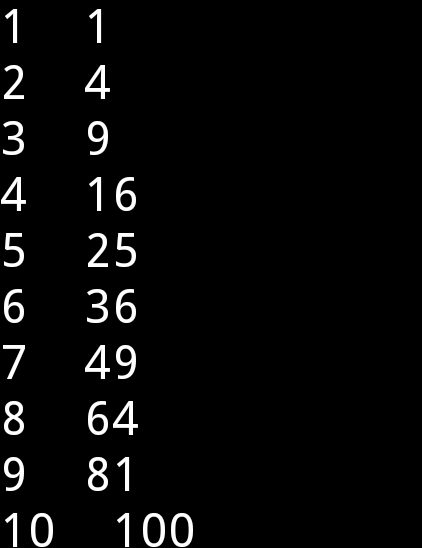
int main() {

for (int nn = 1; nn <= 10; nn++) {

}

return 0;

}



**Практична робота №5**



**Тема: с**творення програм з циклами та розгалуженнями мовою С++.

**Мета**: навчитись складати й реалізовувати циклічні та розгалужені алгоритми та програми мовою С++.

**Завдання №1**

**Варіант №7**

1. Написати програму, яка виводить знаходження перших 7 непарних чисел.

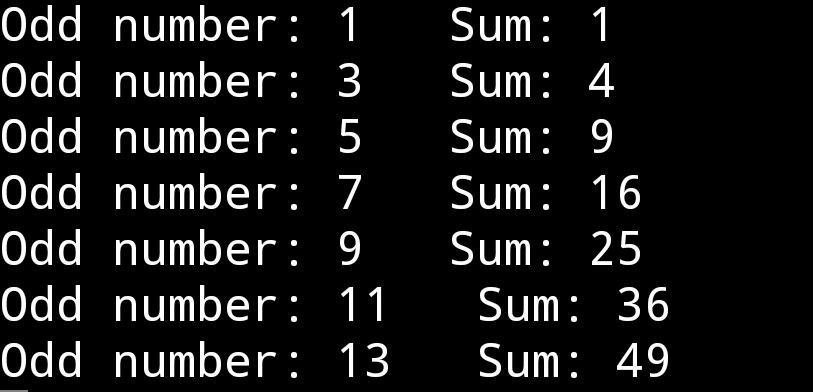
int main() {

for (int sn = 1; sn <= 7; sn++) {

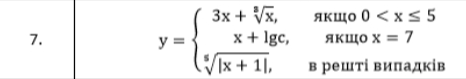
}

return 0;

}



2. Складіть (згідно варіанту) алгоритм програму мовою С++ для знаходження усіх значень у, якщо значення х змінюється від -10 до 10 з кроком ітерації 1. Завдання виконати двома способами організації циклів (FOR і WHILE)



double calculateY(double x) {

if (x > 0 && x <= 5) {

return 3 \* x + pow(x, 1.0 / 3);

}

else if (x == 7) {

return x + log10(x);

}

else {

return pow(abs(x + 1), 1.0 / 5);

}

}

int main()

{

cout << "While loop: \n";

int x = -10;

while (x <= 10) {

x++;

}

cout << "For loop: \n";

for (int x = -10; x <= 10; x++) {

}

return 0;

}

While loop:

x = -10, y = 1.55185

x = -9, y = 1.51572

x = -8, y = 1.47577

x = -7, y = 1.43097



x = -6, y = 1.37973

x = -5, y = 1.31951

x = -4, y = 1.24573

x = -3, y = 1.1487

x = -2, y = 1

x = -1, y = 0

x = 0, y = 1

x = 1, y = 4

x = 2, y = 7.25992

x = 3, y = 10.4422

x = 4, y = 13.5874

x = 5, y = 16.71

x = 6, y = 1.47577

x = 7, y = 7.8451

x = 8, y = 1.55185

x = 9, y = 1.58489

x = 10, y = 1.61539

For loop:

x = -10, y = 1.55185

x = -9, y = 1.51572

x = -8, y = 1.47577

x = -7, y = 1.43097

x = -6, y = 1.37973

x = -5, y = 1.31951

x = -4, y = 1.24573

x = -3, y = 1.1487

x = -2, y = 1

x = -1, y = 0

x = 0, y = 1

x = 1, y = 4

x = 2, y = 7.25992

x = 3, y = 10.4422

x = 4, y = 13.5874

x = 5, y = 16.71

x = 6, y = 1.47577

x = 7, y = 7.8451

x = 8, y = 1.55185

x = 9, y = 1.58489

x = 10, y = 1.61539

**Практична робота №6**

**Тема:** масиви.

**Мета**: оолодіння практичними навичками роботи з масивами.

**Завдання №1**

1. Випробуйте програму 5.1. Зробіть кожен елемент масиву рівним 5.

int main() {

int B[8];

for (int i = 0; i < 8; i++) {

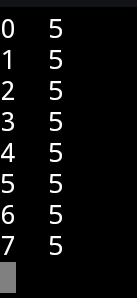
B[i] = 5;

}

return 0;

}





2. Випробуйте програму 5.2. Як змінити програму, щоб останні 4 елементи дорівнювали нулю?

int main() {

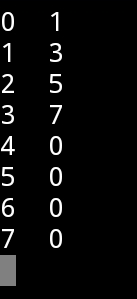
int D[8] = { 1, 3, 5, 7 };

for (int i = 0; i < 8; i++) {

}

return 0;

}



3. Випробуйте програму 5.3. Задайте розмір масиву рівним 15. Виведіть на екран елементи масиву з непарними номерами

int main() {

const int n = 15;

int M[n];

for (int k = 0; k < n; k++) {

M[k] = k + 1;

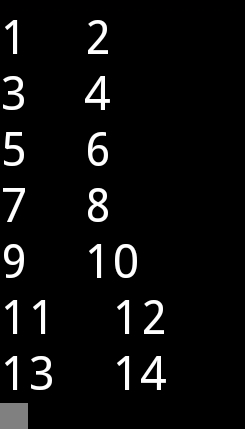
if ((k % 2) != 0) {

}

}

return 0;

}



**Завдання №2**

1. Випробуйте програму 5.4 для n=10 та n=5. Знайдіть добуток всіх елементів таблиці А, яка містить тільки додатні цілі числа.

int main() {

const int r = 10;

int A[r];

for (int j = 0; j < r; j++) {

cout << "A[" << j << "] = ";

cin >> A[j];

if (A[j] <= 0) {



cout << "Error";

return 0;

}

}

int prd = 1;

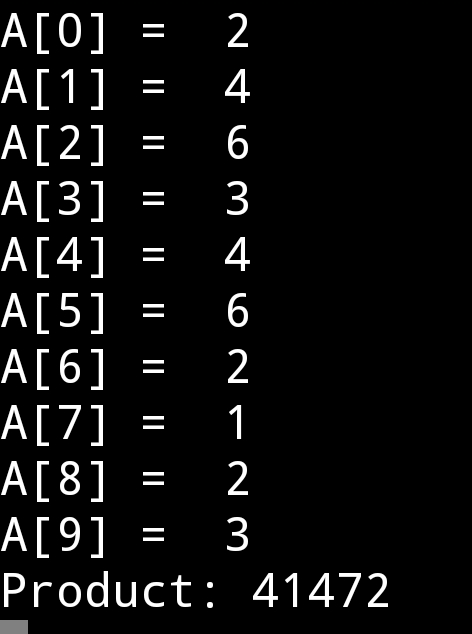
for (int j = 0; j < r; j++) {

prd \*= A[j];

}

return 0;

}



2. Випробуйте програму 5.5. Підрахуйте і надрукуйте кількість додатних елементів масиву цілих чисел х.

int main() {

const int v = 10;

int x[v];

for (int i = 0; i < v; i++) {

cout << i << " Element: ";

cin >> x[i];

}

int pos = 0;

for (int i = 0; i < v; i++) {

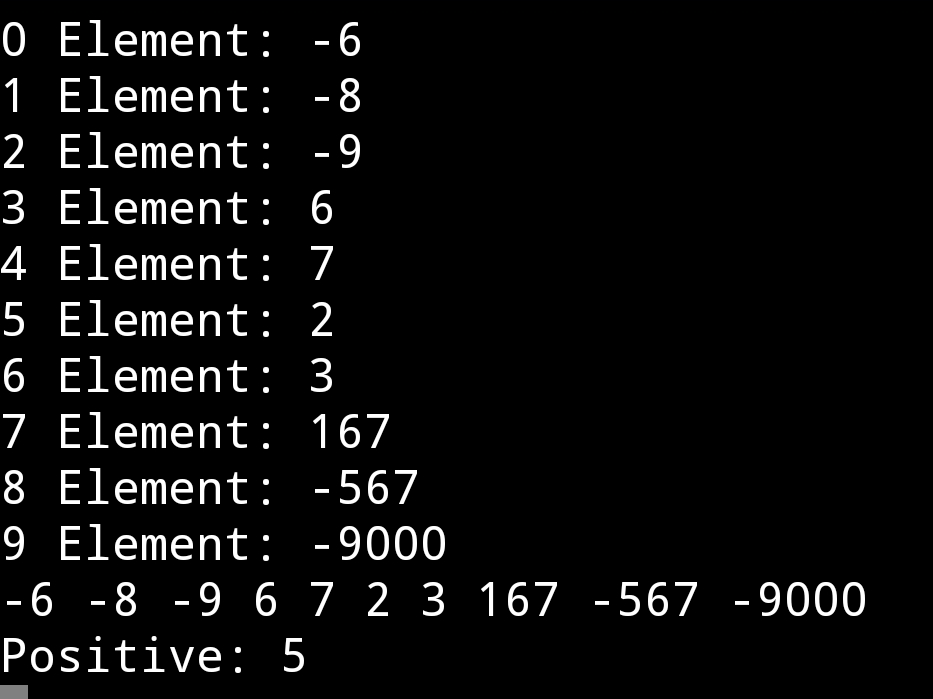
if (x[i] > 0) pos++;

cout << x[i] << " ";

}

return 0;

}



3. Випробуйте програму 5.6. Уважно дослідить роботу програми. Зменшить значення кожного елементу масиву на величину мінімального елемента.

int main() {

const int n = 5;

float a[n];



for (int i = 0; i < n; i++) {

cin >> a[i];

}

float min, max;

min = max = a[0];

for (int i = 1; i < n; i++) {

if (a[i] > max) max = a[i];

if (a[i] < min) min = a[i];

}

for (int i = 0; i < n; i++) {

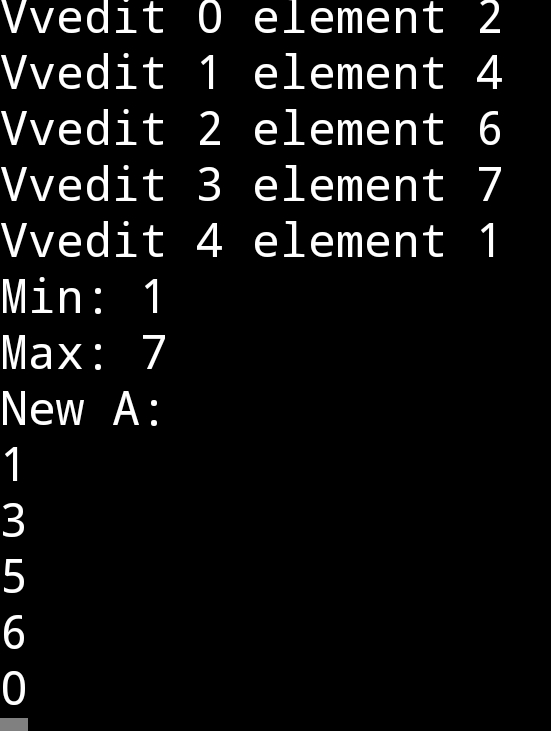
a[i] -= min;

cout << a[i] << endl;

}

return 0;

}



**Практична робота №7**

**Тема:** програмування задач з використаниям двовимірних масивів.

**Мета:** навчитись складати алгоритми задач з двовимірними масивами. Навчитись програмувати задачі з двовимірними масивами.

**Варіант №11**

**Завдання №1**

Заповнити двовимірний масив цілими випалковнми числами в діапазоні від -5 до 5. Вивести елементи масиву по рядках. Розрахувати суму двох будь-яких слементів третього стовпця масиву

int main() {

const int rows = 4;

const int cols = 5;

int array[rows][cols];

srand(time(0));

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

array[i][j] = rand() % 11 - 5;

}

}

cout << "Array:\n";

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {



cout << array[i][j] << "\t";

}

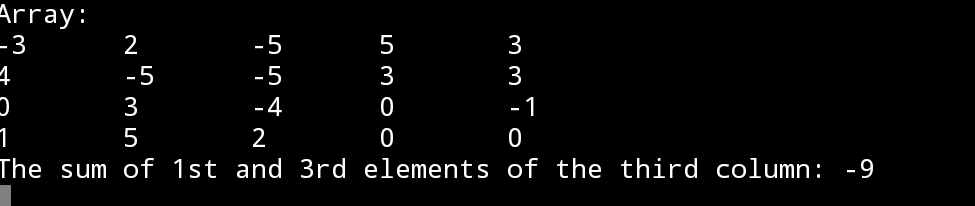
cout << endl;

}

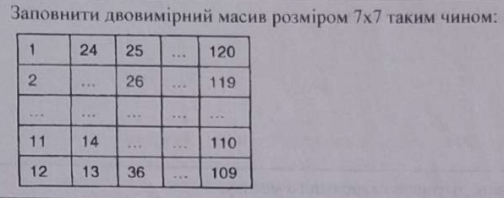
int sum = array[0][2] + array[2][2];

return 0;

}



**Завдання №2**



int main() {

const int rows = 7;

const int cols = 7;

int array[rows][cols];

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

array[i][j] = (j % 2 == 0) ? (12 \* (j + 1) + i - 11) : (12 \* (j + 1) - i);

}

}

cout << "Array:\n";

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

cout << array[i][j] << " ";

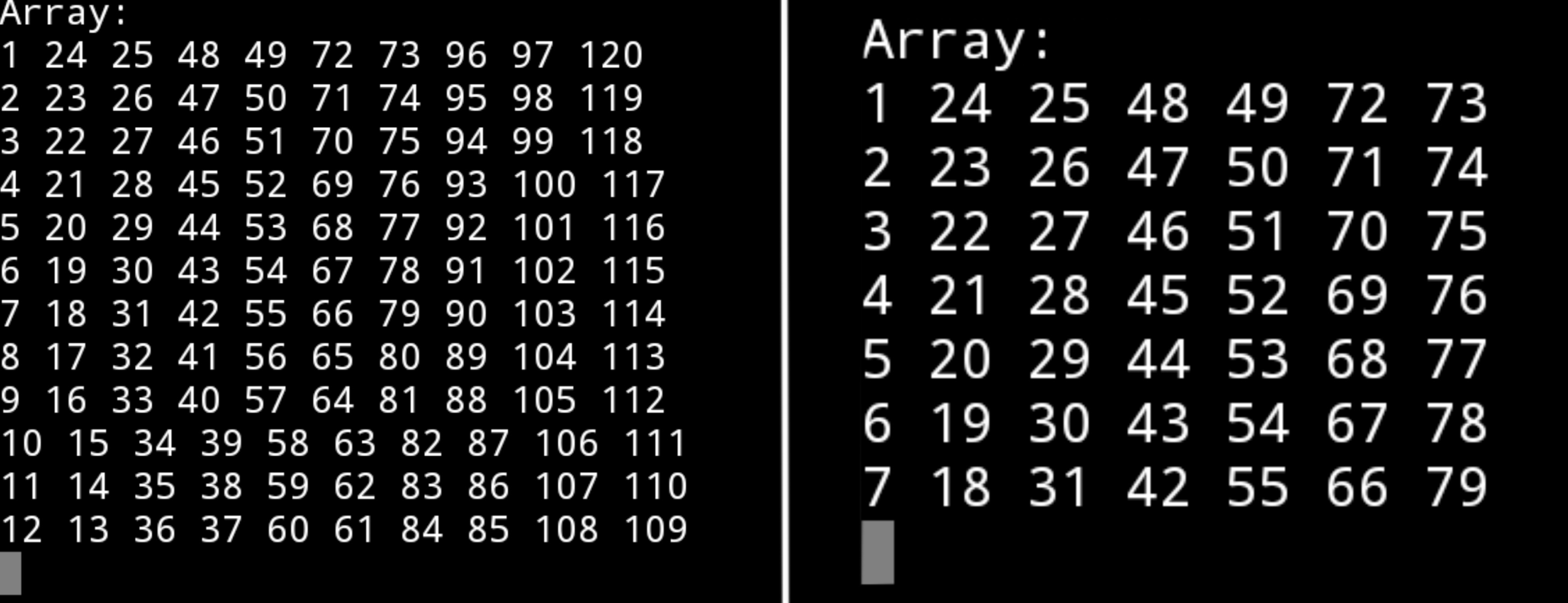
}

cout << endl;

}

return 0;

}





**Завдання №3**

Задано двовиміриий масив з парного числа рядків. Рядки верхньої половини масиву поміияти місцями з рядками нижньої половини

int main() {

srand(time(0));

const int rows = 6;

const int cols = 7;

int array[rows][cols];

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

array[i][j] = rand() % 2000 - 1000;

}

}

cout << "Starting Array:\n";

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

cout << array[i][j] << "\t";

}

cout << endl;

}

// Switch top and bottom halves

for (int i = 0; i < (rows / 2); i++) {

for (int j = 0; j < cols; j++) {

int temp = array[i][j];

array[i][j] = array[rows - i - 1][j];

array[rows - i - 1][j] = temp;

}

}

cout << "Switched Array:\n";

for (int i = 0; i < rows; ++i) {

for (int j = 0; j < cols; ++j) {

cout << array[i][j] << "\t";

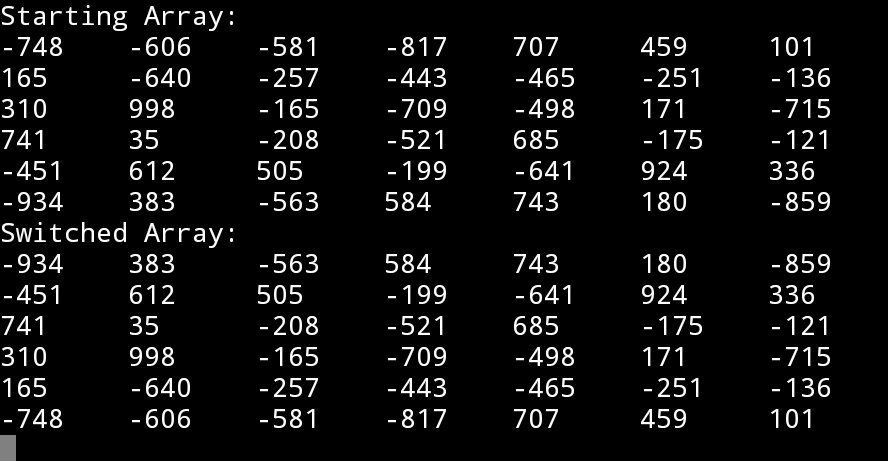
}

cout << endl;

}

return 0;

}



**Практична робота №8**

**Тема:** програмування задач з використаниям рядків.

**Мета:** Навчитись програмувати задачі з використаниям рядків.

**Завдання №1**

**Варіант №13**



Дані натуральне число n, символи s1...sn. Перетворити послідовність s1...n, замінивши в ній всі знаки оклику крапками.

int main() {

int n;

cout << "Enter number of characters: ";

cin >> n;

char\* characters = new char[n];

cout << "Enter characters:\n";

for (int i = 0; i < n; ++i) {

cin >> characters[i];

}

for (int i = 0; i < n; ++i) {

cout << characters[i] << " ";

}

cout << "\nReplaced characters:\n";

for (int i = 0; i < n; ++i) {

if (characters[i] == '!') characters[i] = '.';

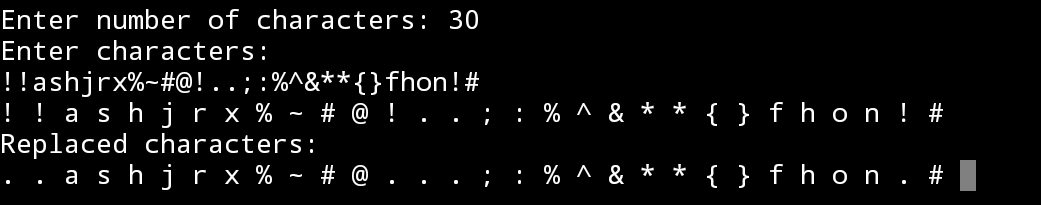
cout << characters[i] << " ";

}

delete[] characters;

return 0;

}



**Завдання №2**

**Варіант №2,**

Визначити скільки букв 'а' є в прізвищі.

int main() {

string lname;

cout << "Enter last name: ";

cin >> lname;

int counta = 0;

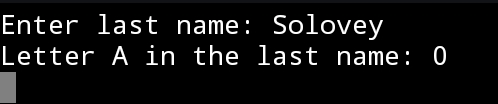
for (char c : lname) {

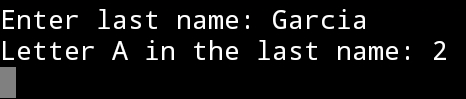
if (c == 'a' || c == 'A') counta++;

}

return 0;

}





**Варіант №7**

Визначити скільки букв 'о' є в імені.

int main() {

string name;

cout << "Enter name: ";

cin >> name;

int counto = 0;

for (char c : name) {

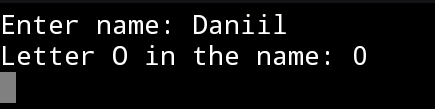
if (c == 'o' || c == 'O') counto++;

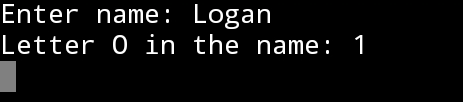
}



return 0;

}





**Завдання №3**

Придумати й описати словесно власний спосіб шифрувания тексту. Скласти програму для введення тексту, як даного тиnу string (до 255 символів), його шифрування. Результати виведіть у файл.

**Завдання №4**

Використовуючи файл, створений під час виконання попередньої задачі, скласти програму дешифрування тексту. Дешифрований текст вивести на екран і у файл.

#include <iostream>

#include <fstream>

#include <string>

#include <cstdlib>

#include <cctype>

using namespace std;

char encryptChar(char c, int shift) {

if (isalpha(c)) {

char base = islower(c) ? 'a' : 'A';

char mirrored = base + ('Z' - toupper(c));

return base + (mirrored - base + shift) % 26;

}

return c;

}

char decryptChar(char c, int shift) {

if (isalpha(c)) {

char base = islower(c) ? 'a' : 'A';

char unshifted = base + (c - base - shift + 26) % 26;

return base + ('Z' - toupper(unshifted));

}

return c;

}

int getShiftFromKey() {

int key;

cout << "Enter integer as key: ";

cin >> key;

srand(key);

return rand() % 25;

}

string processText(const string& text, int shift, bool encryptMode) {

string result;

for (char c : text) {

result += encryptMode ? encryptChar(c, shift) : decryptChar(c, shift);

}

return result;

}

void encrypt() {

string input;

cout << "Enter text to encrypt (max 255 chars): ";



cin >> input;

if (input.length() > 255) input = input.substr(0, 255);

int shift = getShiftFromKey();

string encrypted = processText(input, shift, true);

ofstream outFile("encrypted.txt");

if (outFile.is\_open()) {

outFile << encrypted;

outFile.close();

cout << "Encrypted text saved to encrypted.txt\n";

} else {

cerr << "Error writing to file.\n";

}

}

void decrypt() {

ifstream inFile("encrypted.txt");

string encrypted;

if (inFile.is\_open()) {

getline(inFile, encrypted);

inFile.close();

} else {

cerr << "Error reading encrypted.txt\n";

return;

}

int shift = getShiftFromKey();

string decrypted = processText(encrypted, shift, false);

ofstream outFile("decrypted.txt");

if (outFile.is\_open()) {

outFile << decrypted;

outFile.close();

} else {

cerr << "Error writing decrypted.txt\n";

}

}

int main() {

char c;

cout << "Encrypt(e) or decrypt(d): ";

cin >> c;

switch (c) {

case 'e':

encrypt();

break;

case 'd':

decrypt();

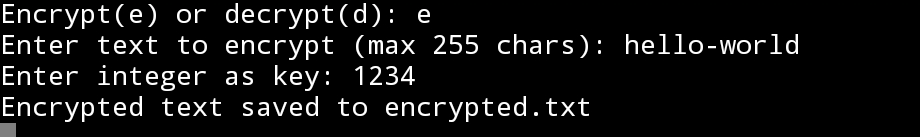
break;

default:

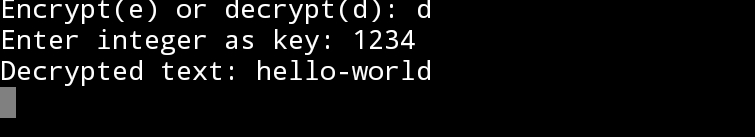
cout << "Error";

}

}







**Практична робота №9**

**Тема:** програмування задач з використанням динамічних масивів.

**Мета:** навчитись програмуати задачі з динамічними масивами.

**Завдання №1**

**Варіант №12**

1. Створіть динамічний масив і заповніть його довільними значеннями.

2. Перевірте, чи є масив впорядкованим за зростанням.

3. Виведіть результат перевірки.

bool isSortedAscending(int arr[], int size) {

for (int i = 1; i < size; ++i) {

if (arr[i] < arr[i - 1]) {

return false;

}

}

return true;

}

int main() {

srand(time(0));

int length = 10;

int \*arr = new int[length];

for (int i = 0; i < length; ++i) {

arr[i] = rand() % 2000 - 1000;

}

cout << "Random array:\n";

for (int i = 0; i < length; ++i) {

cout << arr[i]<<" ";

}

if (isSortedAscending(arr, length)) {

cout << "Sorted\n";

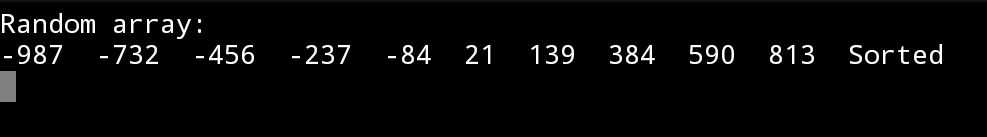
} else {

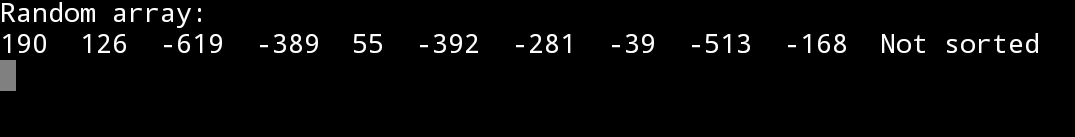
cout << "Not sorted\n";

}

return 0;

}





**Практична робота №10**



**Тема:** програмування задач за допомогою функцій користувача. Перевантаження функцій. Шаблони функцій.

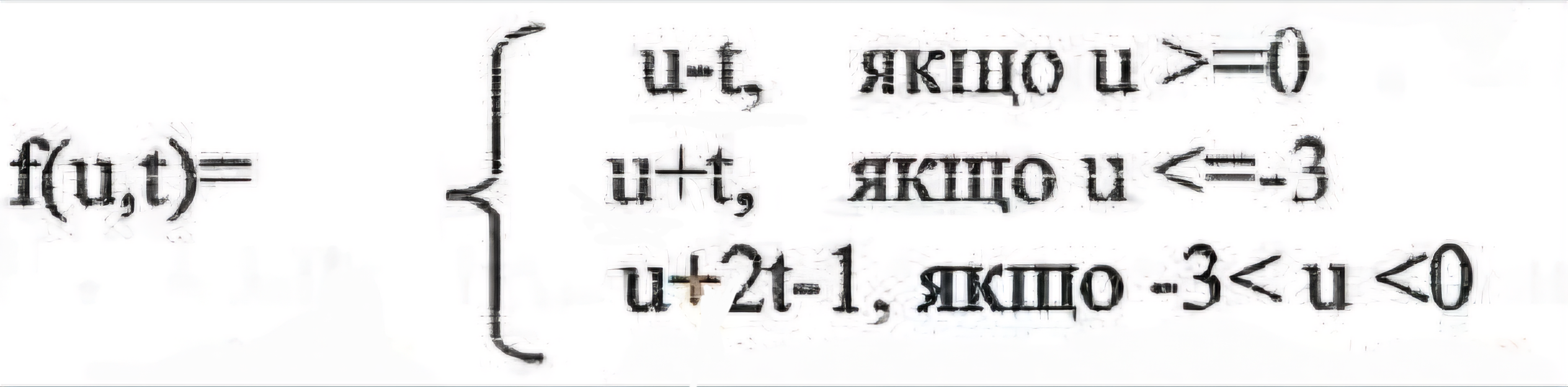
**Мета:** навчитись програмувати задачі з використанням функцій.

**Завдання №1**

**Варіант №7**

Обчислити суму значень функцій.

z=f(x³, y³) + f(a, b²) + f(x⁴ + 1, -y) + f(x² - y³, y) + f(x - y, a - b)



#include <iostream>

#include <cmath>

using namespace std;

double f(double u, double t) {

if (u >= 0) {

return u - t;

} else if (u <= -3) {

return u + t;

} else if (u > -3 && u < 0) {

return u + 2 \* t - 1;

}

}

int main() {

double x, y, a, b;

cout << "Enter x, y, a, b: ";

cin >> x >> y >> a >> b;

double z = 0;

z += f(pow(x, 3), pow(y, 3));

z += f(a, b\*b);

z += f(pow(x, 4) + 1, -y);

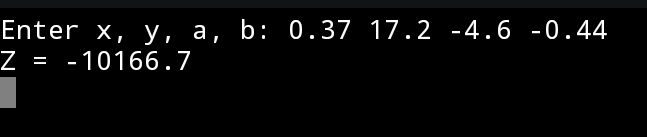
z += f(x\*x - pow(y, 3), y);

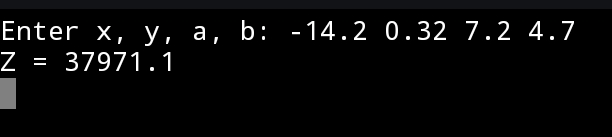
z += f(x - y, a - b);

cout << "Z = "<<z<<endl;

return 0;

}





**Практична робота №11**

**Тема:** програмування задач за допомогою функцій користувача. Перевантаження функцій. Шаблони функцій.

**Мета:** навчитись програмувати задачі з використанням функцій.



**Завдання №1**

**Варіант №20**

Написати програму, яка описує два квадратні масиви дійсного типу розмірністю 2\*3 і 3\*2. Елементи першого масиву необхідно заповнити випадковими числами, другого - задаються при описі. Необхідно обчислити суми квадратів елементів другого рядка першого масиву і першого рядка другого масиву. Результати вивести на екран.

#include <iostream>

#include <cstdlib>

#include <ctime>

#include <cmath>

using namespace std;

int main() {

// Define dimensions

const int rows1 = 2, cols1 = 3;

const int rows2 = 3, cols2 = 2;

// Declare the arrays

double array1[rows1][cols1];

double array2[rows2][cols2] = {

{1.1, 2.2},

{3.3, 4.4},

{5.5, 6.6}

};

// Seed the random number generator

srand(time(0));

// Fill the first array with random real numbers between 0 and 100

cout << "Array 1:" << endl;

for (int i = 0; i < rows1; i++) {

for (int j = 0; j < cols1; j++) {

array1[i][j] = rand() % 100;

cout << array1[i][j] << " ";

}

cout << endl;

}

// Display second array

cout << "\nArray 2:" << endl;

for (int i = 0; i < rows2; i++) {

for (int j = 0; j < cols2; j++) {

cout << array2[i][j] << " ";

}

cout << endl;

}

// Calculate sum of squares of the second row of the first array

double sum1 = 0;

for (int j = 0; j < cols1; j++) {

sum1 += pow(array1[1][j], 2);

}

// Calculate sum of squares of the first row of the second array

double sum2 = 0;

for (int j = 0; j < cols2; j++) {

sum2 += pow(array2[0][j], 2);

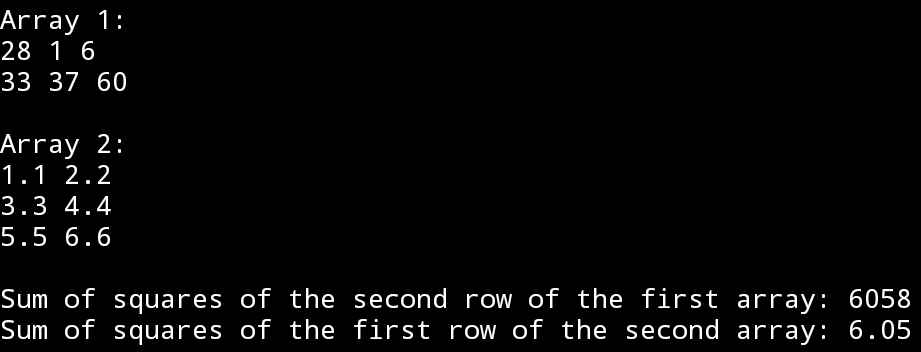
}

// Output results



return 0;

}



**Практична робота №12**

**Тема:** програмування задач з використанням текстових файлів.

**Мета:** навчитись програмуати задачі з послідовним доступом до зовнішніх носіїв.

**Завдання №1**

**Варіант №12**

a) виводить кількість слів, які містять однакову кількість голосних приголосних, літер;

б) виводить на екран найдовше слово;

в) видаляє тексту всі слова-паліндроми.

#include <iostream>

#include <fstream>

#include <vector>

using namespace std;

bool isVowel(char c) {

c = tolower(c);

return (c == 'a' || c == 'e' || c == 'i' || c == 'o' || c == 'u');

}

bool isPalindrome(const string &word) {

string cleaned = "";

for (int i = 0; i < word.length(); i++) {

if (isalpha(word[i])) {

cleaned += tolower(word[i]);

}

}

int n = cleaned.length();

for (int i = 0; i < n / 2; i++) {

if (cleaned[i] != cleaned[n - 1 - i])

return false;

}

return n > 0;

}

int main() {



ifstream infile("input.txt");

ofstream outfile("output.txt");

if (!infile.is\_open() || !outfile.is\_open()) {

cerr << "Error opening input or output file." << endl;

return 1;

}

string word;

vector<string> words;

int equalCount = 0;

string longest = "";

while (infile >> word) {

int vowels = 0, consonants = 0;

for (int i = 0; i < word.length(); i++) {

if (isalpha(word[i])) {

if (isVowel(word[i]))

vowels++;

else

consonants++;

}

}

if (vowels == consonants && (vowels + consonants) > 0)

equalCount++;

if (word.length() > longest.length())

longest = word;

if (!isPalindrome(word))

words.push\_back(word);

}

// a) Output count

// b) Longest word

// c) Words without palindromes

cout << "Text without palindromic words:" << endl;

outfile << "Text without palindromic words:" << endl;

for (int i = 0; i < words.size(); i++) {

cout << words[i] << " ";

outfile << words[i] << " ";

}

cout << endl;

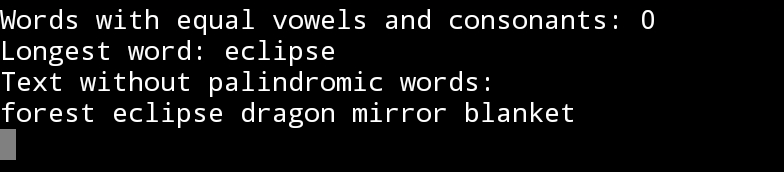
outfile << endl;

infile.close();

outfile.close();

return 0;

}





**Практична робота №13, 14, 15**

**Тема:** програмування задач з використанням структур.

**Мета:** навчитись програмувати задачі з структурами.

**Завдання №1**

**Варіант №1**

Написати програму, яка виконує наступні дії:

* Описує структуру з ім'ям STUDENT, що містить такі поля:
* прізвище та ініціали;
* номер групи;
* успішність по п’яти предметах (масив з п’яти елементів).
* Здійснює введення з клавіатури даних у масив структур, що складається з 5 елементів.
* Записує дані з масиву структур у файл.
* Зчитує дані з файлу в масив структур і виводить їх на екран.
* Виводить на дисплей прізвища та номери груп для всіх студентів, у яких середній бал більше 4.0. Якщо таких студентів немає, вивести відповідне повідомлення.

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

struct STUDENT {

string lastName;

string initials;

string groupNumber;

int grades[5];

};

const int NUM\_STUDENTS = 5;

// Function to calculate average grade

double averageGrade(const STUDENT& student) {

int sum = 0;

for (int i = 0; i < 5; ++i) {

sum += student.grades[i];

}

return sum / 5.0;

}

int main() {

STUDENT students[NUM\_STUDENTS];

// Input data from keyboard

cout << "Enter data for 5 students:\n";

for (int i = 0; i < NUM\_STUDENTS; ++i) {

cout << "Last name: ";

cin.ignore();

getline(cin, students[i].lastName);

cout << "Initials: ";

getline(cin, students[i].initials);

cout << "Group number: ";

getline(cin, students[i].groupNumber);

cout << "Enter 5 grades: ";

for (int j = 0; j < 5; ++j) {

cin >> students[i].grades[j];

}

}

// Write to file



ofstream outFile("students.txt");

for (int i = 0; i < NUM\_STUDENTS; ++i) {

outFile << students[i].lastName << "\n"

<< students[i].initials << "\n"

<< students[i].groupNumber << "\n";

for (int j = 0; j < 5; ++j) {

outFile << students[i].grades[j] << " ";

}

outFile << "\n";

}

outFile.close();

// Read from file

ifstream inFile("students.txt");

STUDENT readStudents[NUM\_STUDENTS];

for (int i = 0; i < NUM\_STUDENTS; ++i) {

getline(inFile, readStudents[i].lastName);

getline(inFile, readStudents[i].initials);

getline(inFile, readStudents[i].groupNumber);

for (int j = 0; j < 5; ++j) {

inFile >> readStudents[i].grades[j];

}

inFile.ignore(); // skip newline after grades

}

inFile.close();

// Display students with average > 4.0

bool found = false;

cout << "\nStudents with average grade > 4.0:\n";

for (int i = 0; i < NUM\_STUDENTS; ++i) {

if (averageGrade(readStudents[i]) > 4.0) {

found = true;

}

}

if (!found) {

cout << "No students with average grade > 4.0\n";

}

return 0;

}

