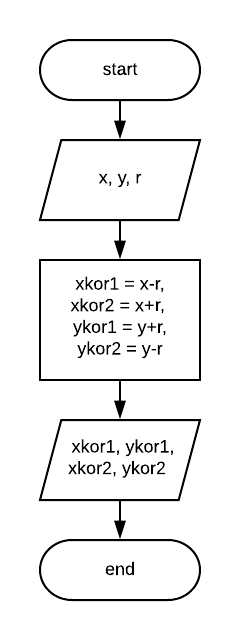
Задача 1 (лінійні алгоритми)

Постановка задачі:

Визначити координати проекцій кола із центром у точці (х;у) та радіусом r на осі координат.

Блок-схема алгоритму:

В даній задачі користувач вводить точки центра кола та його радіус з клавіатури. Після чого програма обчислює проекції, додавши чи віднявши радіус від координат х та у центру кола. Розв’язок виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251);// підключення української мови

// оголошуємо змінні

float x, y, r;

// ввід данних користувачем

cout << "Введіть координати центру кола та його радіус: \n";

cout << "x = "; cin >> x; cout << "y = "; cin >> y;

cout << "r = "; cin >> r;

float xkor1 = x-r , xkor2 = x+r, ykor1 = y+r, ykor2= y-r;

// обчислення і вивід результатів

cout << "Проекції кола на ось координат:\n";

cout << "x1 = (" << xkor1 << " ; 0)\n";

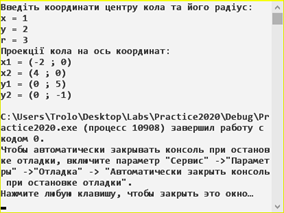
cout << "x2 = (" << xkor2 << " ; 0)\n";

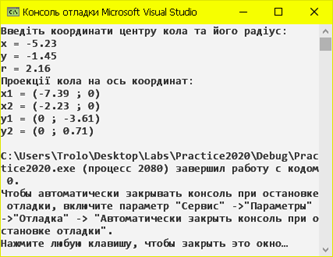
cout << "y1 = (0 ; " << ykor1 << ")\n";

cout << "y2 = (0 ; " << ykor2 << ")\n";

}

Результат виконання:





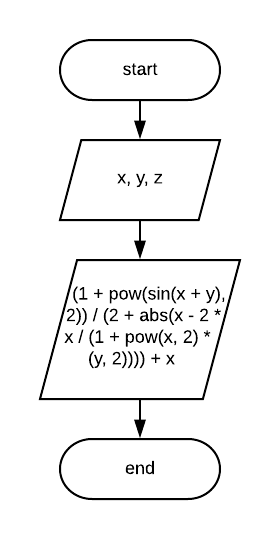
Задача 2 (лінійні алгоритми)

Постановка задачі:

Дано змінні x, y, z. Обчислити значення змінних a або b.

Блок-схема алгоритму:

В даній програмі користувач вводить x, y та z з клавіатури. Після чого програма обчислює значення а та виводить результат на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

float x, y, z;

// обчислення і вивід результатів

cout << "Введіть числа: \n";

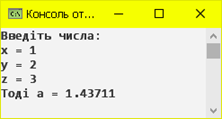
cout << "x = "; cin >> x; cout << "y = "; cin >> y;

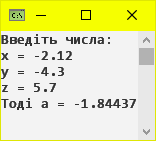
cout << "z = "; cin >> z;

cout << "Тоді а = " << (1 + pow(sin(x + y), 2)) / (2 + abs(x - 2 \* x / (1 + pow(x, 2) \* (y, 2)))) + x << "\n";

}

Результат виконання:





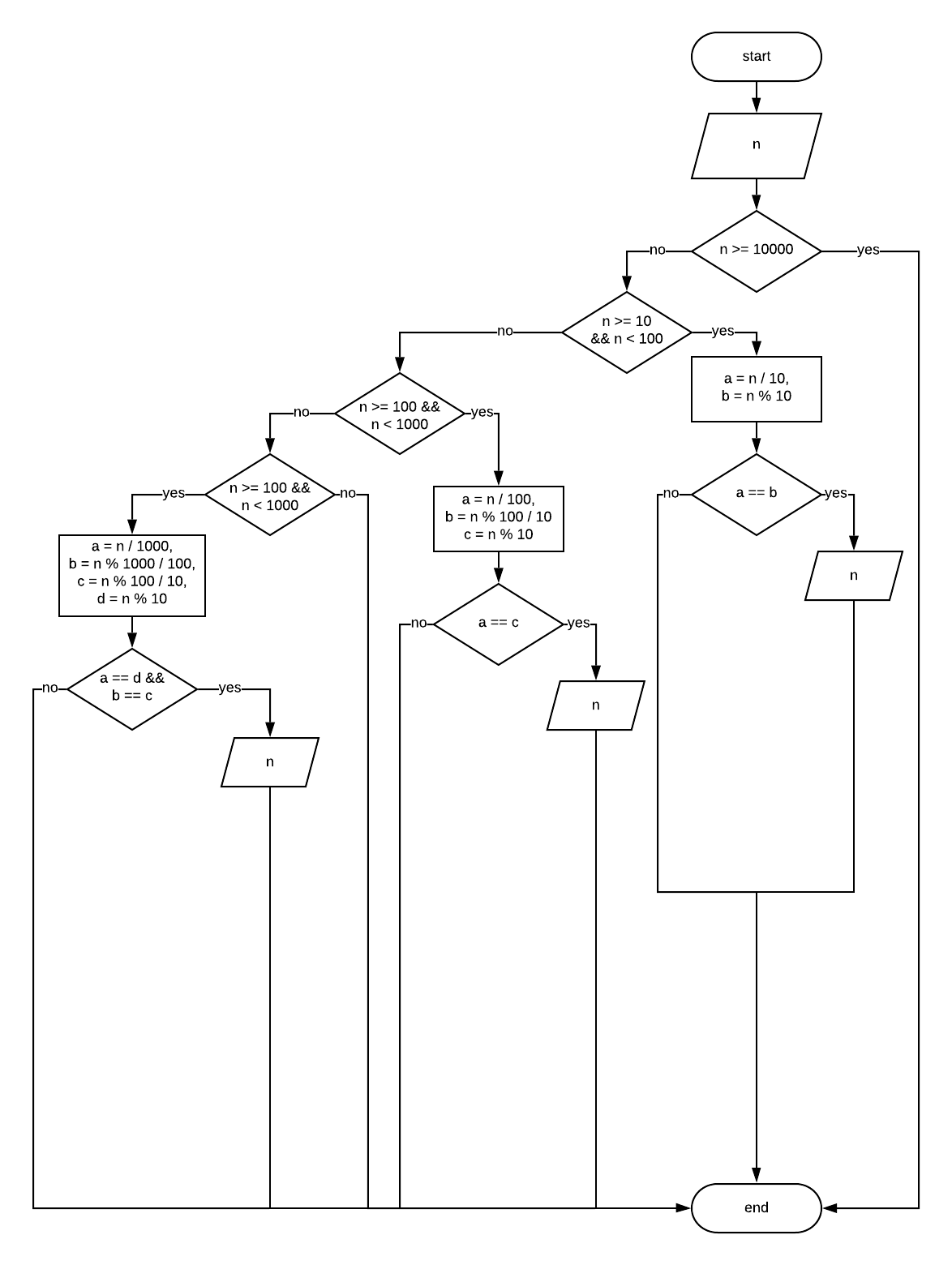
Задача 3 (розгалужені алгоритми)

Постановка задачі:

Дано натуральне число n (n<10000). Визначити, чи є це число паліндромом, тобто таким числом, яке зліва направо і справа наліво читається однаково (всі чотири цивфри враховуються; наприклад, 0110).

Блок-схема алгоритму:

В даній задачі користувач вводить число n з клавіатури. Після чого програма визначає, скільки число має чисел. Далі враховуючи кількість розбиває його на пари і визначає, чи це паліндром. Відповідь виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

int n;

// ввід данних користувачем

cout << "Введіть число n = "; cin >> n;

// обчислення і вивід результатів

if (n >= 10000) { cout << "Помилка, n має бути меншим за 10000\n"; return(0); }

else if (n >= 10 && n < 100)

{

int a = n / 10;

int b = n % 10;

if (a == b) cout << "Число є паліндромом" << endl;

else cout << "Число НЕ є паліндромом" << endl;

}

else if (n >= 100 && n < 1000)

{

int a = n / 100;

int b = n % 100 / 10;

int c = n % 10;

if (a == c) cout << "Число є паліндромом" << endl;

else cout << "Число НЕ є паліндромом" << endl;

}

else if (n>=1000 && n<10000)

{

int a = n / 1000;

int b = n % 1000 / 100;

int c = n % 100 / 10;

int d = n % 10;

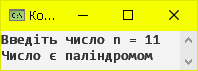
if (a == d && b == c) cout << "Число є паліндромом" << endl;

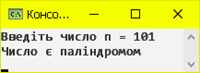
else cout << "Число НЕ є паліндромом" << endl;

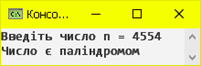
}

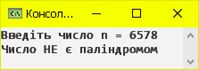
}

Результат виконання:





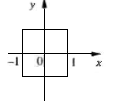




Задача 4 (розгалужені алгоритми)

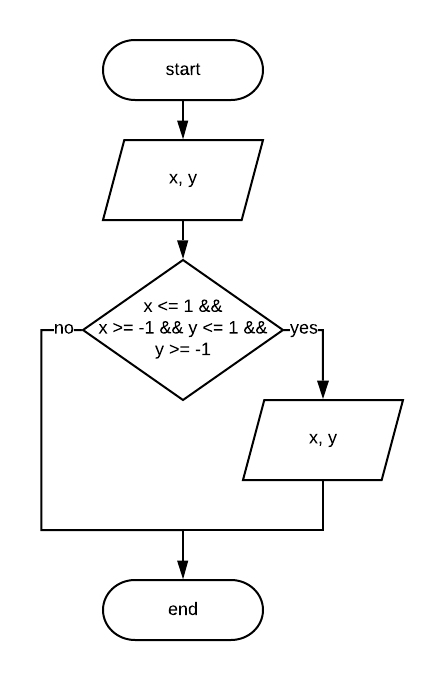
Постановка задачі:

Дано дійсні числа х, у. Визначити, чи належить точка з координатами (х;у) зафарбованій частині площини.



Блок-схема алгоритму:

В даній задачі користувач вводить числа х та у з клавіатури. Після програма перевіряє, чи належить точка з заданими координатами зафарбованій частині площини. Відповідь виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

float x, y;

cout << "Введіть координати точки: \n";

// ввід данних користувачем

cout << "x = "; cin >> x; cout << "y = "; cin >> y;

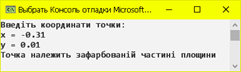
// обчислення і вивід результатів

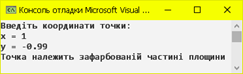
if (x <= 1 && x >= -1 && y <= 1 && y >= -1) cout << "Точка належить зафарбованій частині площини\n";

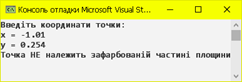
else cout << "Точка НЕ належить зафарбованій частині площини\n";

}

Результат виконання:







Задача 5 (циклічні алгоритми)

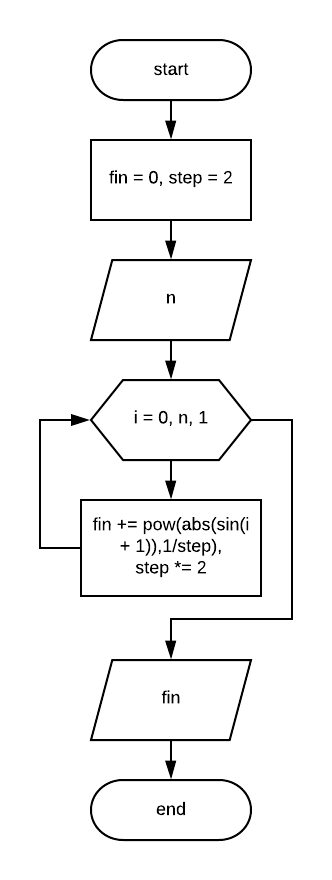
Постановка задачі:

За заданим натуральним значенням змінної n обчислити:



Блок-схема алгоритму:

В даній задачі користувач вводить значення n з клавіатури. Після чого програма обчислює число fin за заданою формулою. Відповідь виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

int n;

float fin=0, step = 2;

// ввід данних користувачем

cout << "Введіть число n = "; cin >> n;

// обчислення і вивід результатів

for (float i = 0; i < n; i++)

{

fin += pow(abs(sin(i + 1)),1/step);

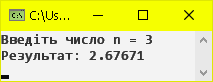
step \*= 2;

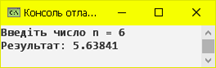
}

cout << "Результат: " << fin << endl;

}

Результат виконання:





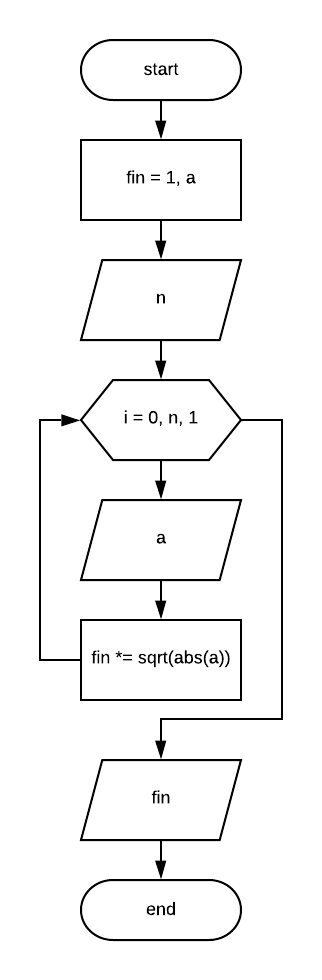
Задача 6 (циклічні алгоритми)

Постановка задачі:

Дано натуральне число n і дійсні числа . Обчислити:

Блок-схема алгоритму:

В даній задачі користувач задає n з клавіатури. Після цикл працює n разів. Користувач має задавати а значеннями, для підрахунку їх добутків за формулою. Далі відповідь виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

int n;

float a, fin=1;

// ввід данних користувачем

cout << "Введіть число n = "; cin >> n;

// обчислення і вивід результатів

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

cout << "a" << i + 1 << " = ";cin >> a;

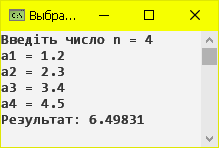
fin \*= sqrt(abs(a));

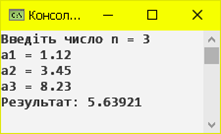
}

cout << "Результат: " << fin << endl;

}

Результат виконання:





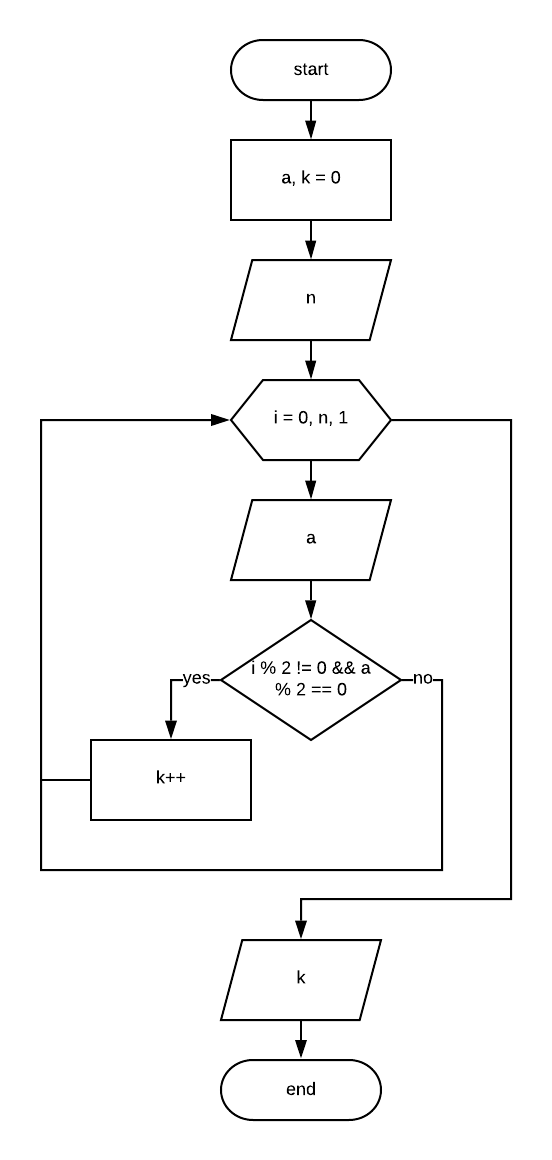
Задача 7 (циклічні алгоритми)

Постановка задачі:

Дано натуральні числа n і . Визначити кількість членів послідовності , що мають непарні порядкові номери та є парними числами.

Блок-схема алгоритму:

В даній задачі користувач вводить число n з клавіатури. Цикл працює n разів. Далі користувач задає числа а. Воно проходить перевірку разом з своїм порядковим номером. При правильному відповіді число k збільшується на 1. Далі відповідь виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

int n;

int a, k = 0;

// ввід данних користувачем

cout << "Введіть число n = "; cin >> n;

// обчислення і вивід результатів

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

cout << "a" << i << " = "; cin >> a;

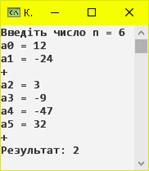
if (i % 2 != 0 && a % 2 == 0) { k++; cout << "+\n"; }

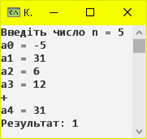
}

cout << "Результат: " << k << endl;

}

Результат виконання:





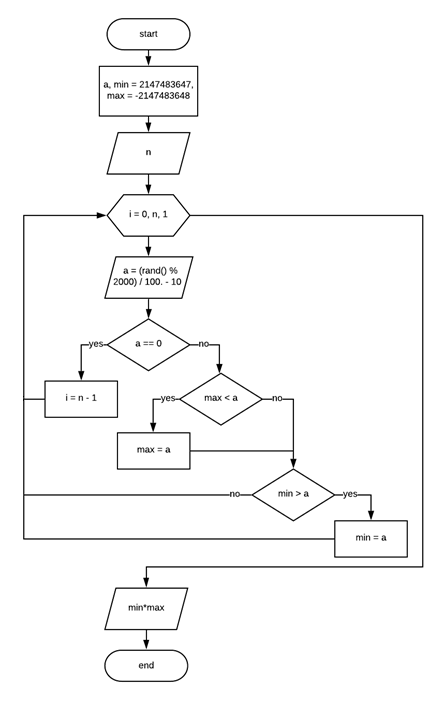
Задача 8 (циклічні алгоритми)

Постановка задачі:

Дано натуральне число n та дійсні числа . Відомо, що члени послідовності що розташовані до першого нульового члена, визначити добуток найбільшого та найменшого членів.

Блок-схема алгоритму:

В даній задачі користувач вводить число n з клавіатури. Цикл працює n разів. Далі масив а заповнюється випадковими числами. Числа проходять перевірку, програма знаходить мінімум і максимум серед цих чисел. Далі їх добуток виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

srand(time(NULL));

// оголошуємо змінні

int n;

double a, min =2147483647, max = -2147483648;

cout << "Введіть число n = "; cin >> n;

// обчислення і вивід результатів

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

a = (rand() % 2000) / 100. - 10;

cout << a << " | ";

if (a == 0) i = n - 1;

if (max < a) max = a;

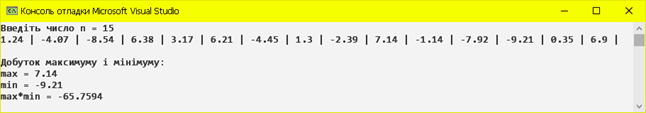
if (min > a) min = a;

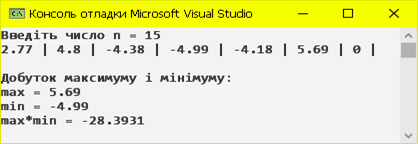
}

cout << "\n\nДобуток максимуму і мінімуму: "<<"\nmax = "<<max<<"\nmin = "<<min<<"\nmax\*min = "<< max\*min << endl;

}

Результат виконання:





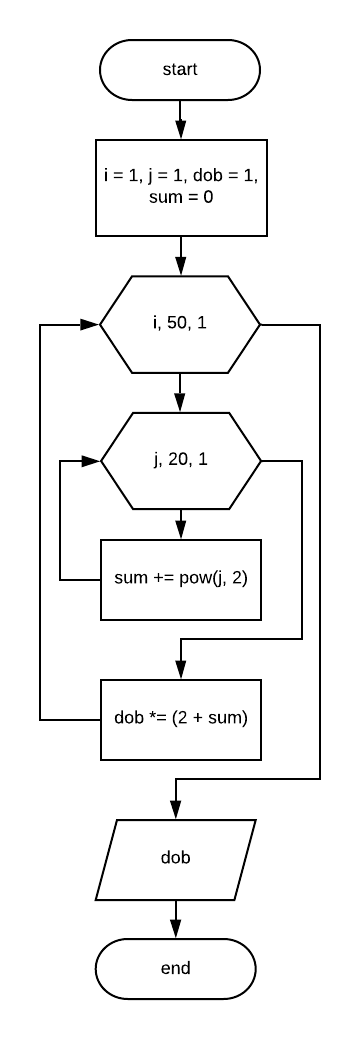
Задача 9 (вкладені циклічні алгоритми)

Постановка задачі:

Скласти програму обчислення:

Блок-схема алгоритму:

В даній задачі вираховується добуток суми чисел і та j. Програма розраховує їх за формулою, зазначеною у дано. Після чого виводить результат на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

int i = 1, j = 1;

double dob = 1, sum = 0;

// обчислення і вивід результатів

for ( i ; i <= 50; i++) {

for (j; j <= 20; j++) {

sum += pow(j, 2);

}

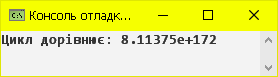
dob \*= (2 + sum);

}

cout << "Цикл дорівнює: " << dob<<endl;

}

Результат виконання:



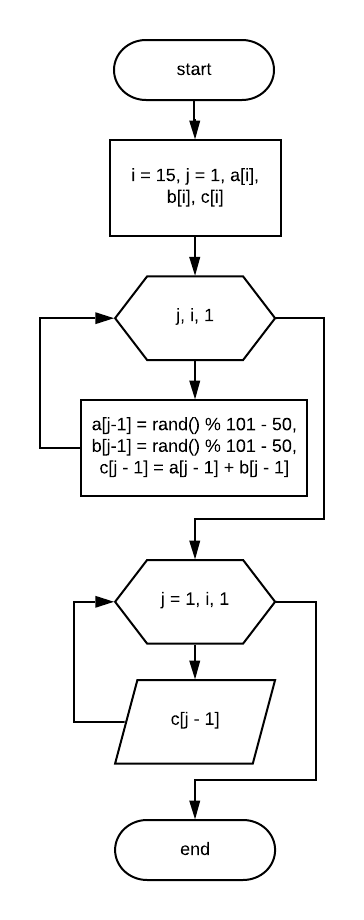
Задача 10 (одновимірні числові масиви)

Постановка задачі:

Дано цілі числа та . Визначити значення якщо:

Блок-схема алгоритму:

В даній задачі масиви а та b наповнюються випадковими числами. Після програма рахує значення чисел для заповнення масиву с за дано. Після масив с виводиться на екран.



Лістинг програми:

int main()

{

srand(time(NULL));

// оголошуємо змінні

const int i = 15;

int j = 1, a[i], b[i], c[i];

// обчислення

for (j; j <= i; j++) {

a[j-1] = rand() % 101 - 50;

cout << "a[" << j << "] = " << a[j - 1] << endl;

b[j-1] = rand() % 101 - 50;

cout << "b[" << j << "] = " << b[j - 1] << endl << endl;

c[j - 1] = a[j - 1] + b[j - 1];

}

// вивід результатів

cout << "c[i] = a[i] + b[i]\n\n";

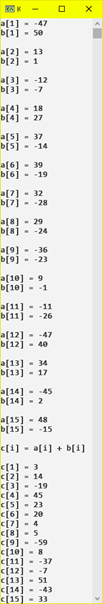
for (j = 1; j <= i; j++) {

cout << "c[" << j << "] = " << c[j - 1] << endl;

}

}

Результат виконання:



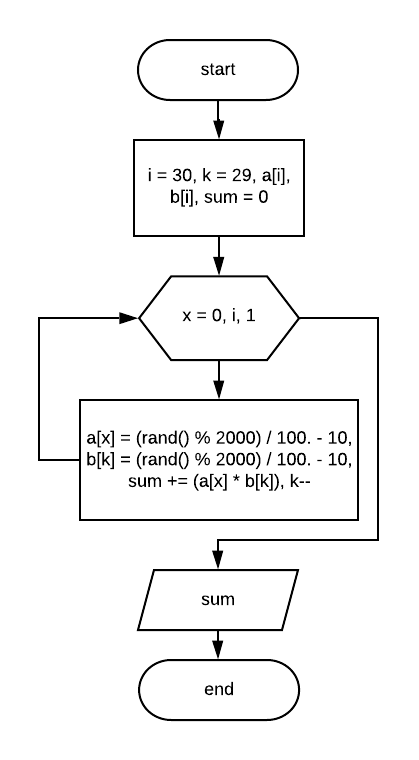
Задача 11 (одновимірні числові масиви)

Постановка задачі:

Дано дійсні числа та . Обчислити:

Блок-схема алгоритму:

В даній задачі масиви а та b наповнюються випадковими числами. Далі програма рахує суму за дано. Після чого відповідь виводиться на екран.



Лістинг програми:

int main()

{

// оголошуємо змінні

srand(time(NULL));

const int i = 30;

int k = 29;

double a[i], b[i], sum = 0;

// обчислення

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

a[x] = (rand() % 2000) / 100. - 10;

cout << "a[" << x+1 << "] = " << a[x] << endl;

b[k] = (rand() % 2000) / 100. - 10;

cout << "b[" << k+1 << "] = " << b[k] << endl << endl;

sum += (a[x] \* b[k]);

k--;

}

// вивід результатів

cout << "sum (a[1]\*b[30]) + (a[2]\*b[29])...(a[30]\*b[1]) = " << sum;

}

Результат виконання:



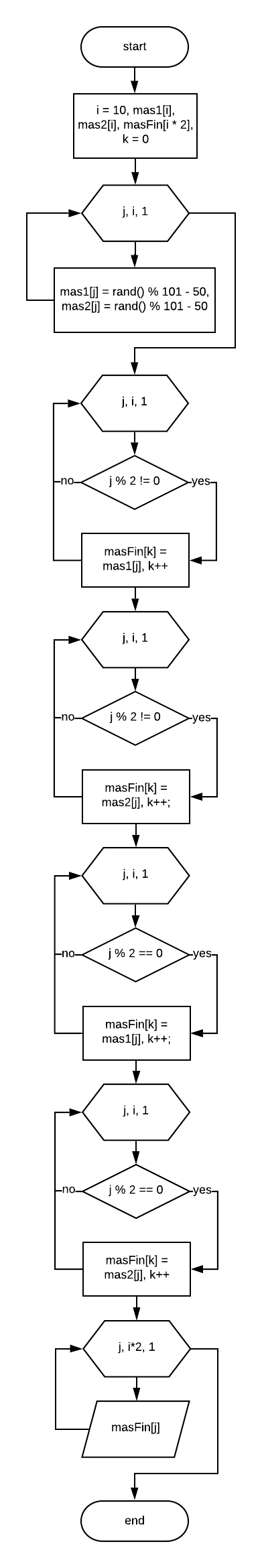
Задача 12 (одновимірні числові масиви)

Постановка задачі:

Дано два одновимірних масиви. Утворити новий масив, у якому спочатку будуть розташовані всі парні елементи першого масиву, потім парні елементи другого, далі непарні елементи першого і непарні елементи другого масиву.

Блок-схема алгоритму:

В даній задачі два одновимірних масиви наповнюються випадковими числами. Далі, щоб наповнити новий масив, виконуються 4 схожих цикли. За дано, спочатку наповнюється всі парні елементи першого масиву, потім парні елементи другого, далі непарні елементи першого і непарні елементи другого масиву. Після чого новий масив виводиться на екран.



Лістинг програми:

int main()

{

srand(time(NULL));

// оголошуємо змінні

const int i = 10;

int mas1[i], mas2[i], masFin[i \* 2], k = 0;

cout <<"mas1 = { ";

// обчислення

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

mas1[j] = rand() % 101 - 50;

cout << mas1[j]<< " ";

mas2[j] = rand() % 101 - 50;

}

cout << "}\n" << "mas2 = { ";

for (int j = 0; j < i; j++)

cout << mas2[j] << " ";

cout << "}\n";

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

if (j % 2 != 0) {

masFin[k] = mas1[j];

k++;

}

}

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

if (j % 2 != 0) {

masFin[k] = mas2[j];

k++;

}

}

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

if (j % 2 == 0) {

masFin[k] = mas1[j];

k++;

}

}

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

if (j % 2 == 0) {

masFin[k] = mas2[j];

k++;

}

}

cout <<"\nmas3 = { ";

// вивід результатів

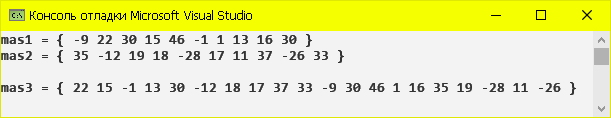
for (int j = 0; j < (i\*2); j++)

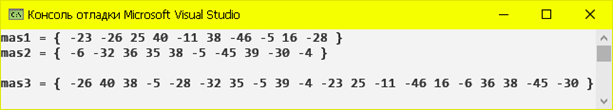
cout << masFin[j] << " ";

cout << "}\n";

}

Результат виконання:





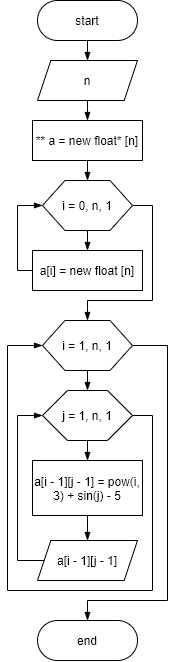
Задача 13 (двовимірні числові масиви)

Постановка задачі:

Дано натуральне число n. Обчислити елементи квадратної матриці за такою формулою:

Блок-схема алгоритму:

В даній програмі користувач вводить значення n з клавіатури. Після чого створюється двовимірна матриця з розмірністю n на n. Далі програма рахує елементи матриці за формулою із дано. Вирішена матриця виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

srand(time(NULL));

// оголошуємо змінні

int n;

// ввід данних користувачем

cout << "Введіть розмірність матриці: n = "; cin >> n;

cout << "Матриця:\n";

float\*\* a = new float\* [n];

for (int i = 0; i < n; i++) a[i] = new float[n];

// обчислення і вивід результатів

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

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

a[i - 1][j - 1] = pow(i, 3) + sin(j) - 5;

cout << setw(9) << a[i - 1][j - 1];

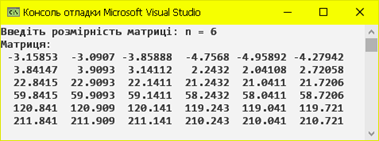
}

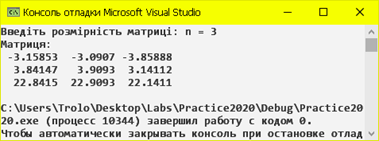
cout << endl;

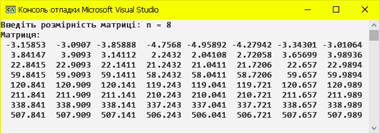
}

}

Результат виконання:







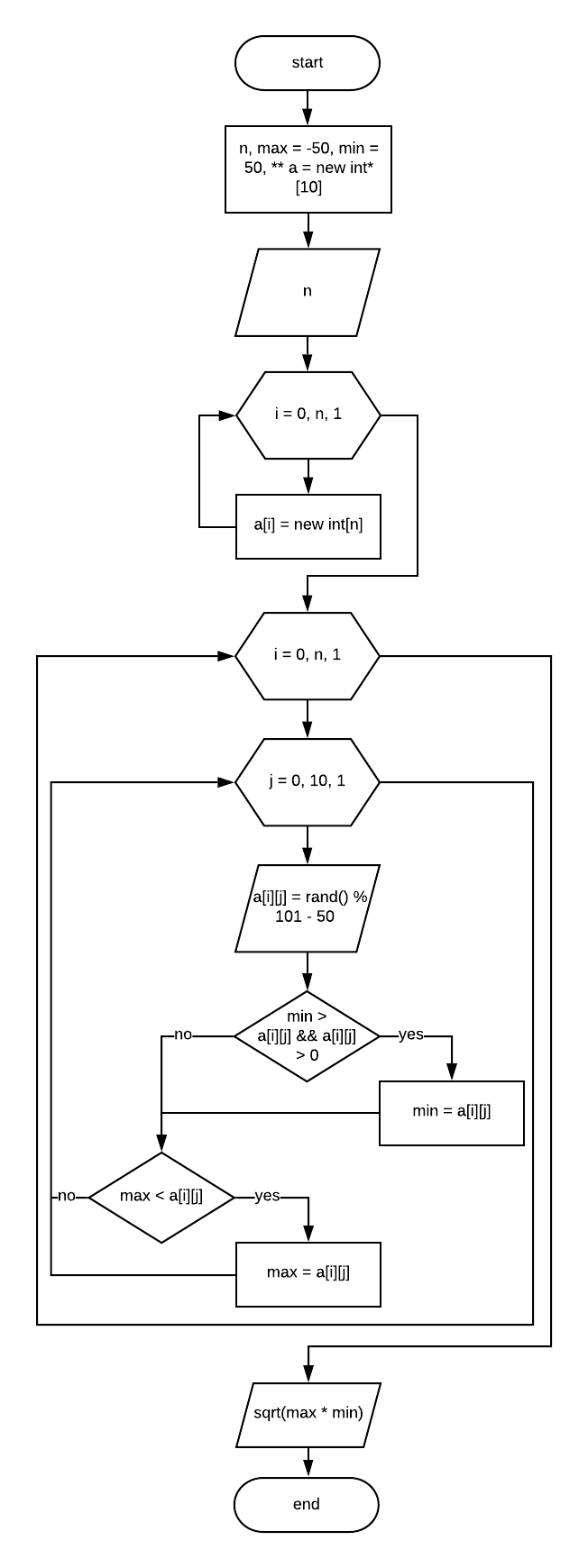
Задача 14 (двовимірні числові масиви)

Постановка задачі:

Дано прямокутну цілочисельну матрицю порядку . Визначити середнє геометричне максимального та мінімального серед додатних значень.

Блок-схема алгоритму:

В даній задачі користувач вводить значення n з клавіатури. Програма наповнює динамічний масив випадковими значеннями. Далі знаходимо мінімальний та максимальний додатні елементи масиву. Після програма виводить добуток їх кореню на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

srand(time(NULL));

// оголошуємо змінні

int n;

float max = -50, min = 50;

// ввід данних користувачем

cout << "Введіть к-сть стовпців матриці: "; cin >> n;

cout << "Матриця:\n";

// обчислення

int\*\* a = new int\* [10];

for (int i = 0; i < 10; i++) a[i] = new int[n];

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

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

a[i][j] = rand() % 101 - 50;

cout.width(4);

cout << a[i][j];

if (min > a[i][j] && a[i][j] > 0) min = a[i][j];

if (max < a[i][j]) max = a[i][j];

}

cout << endl;

}

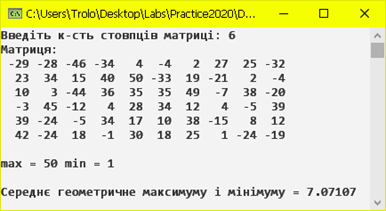
// вивід результатів

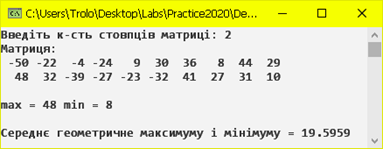
cout << "\nmax = " << max << " min = " << min << endl;

cout << "\nСереднє геометричне максимуму і мінімуму = " << sqrt(max \* min);

}

Результат виконання:





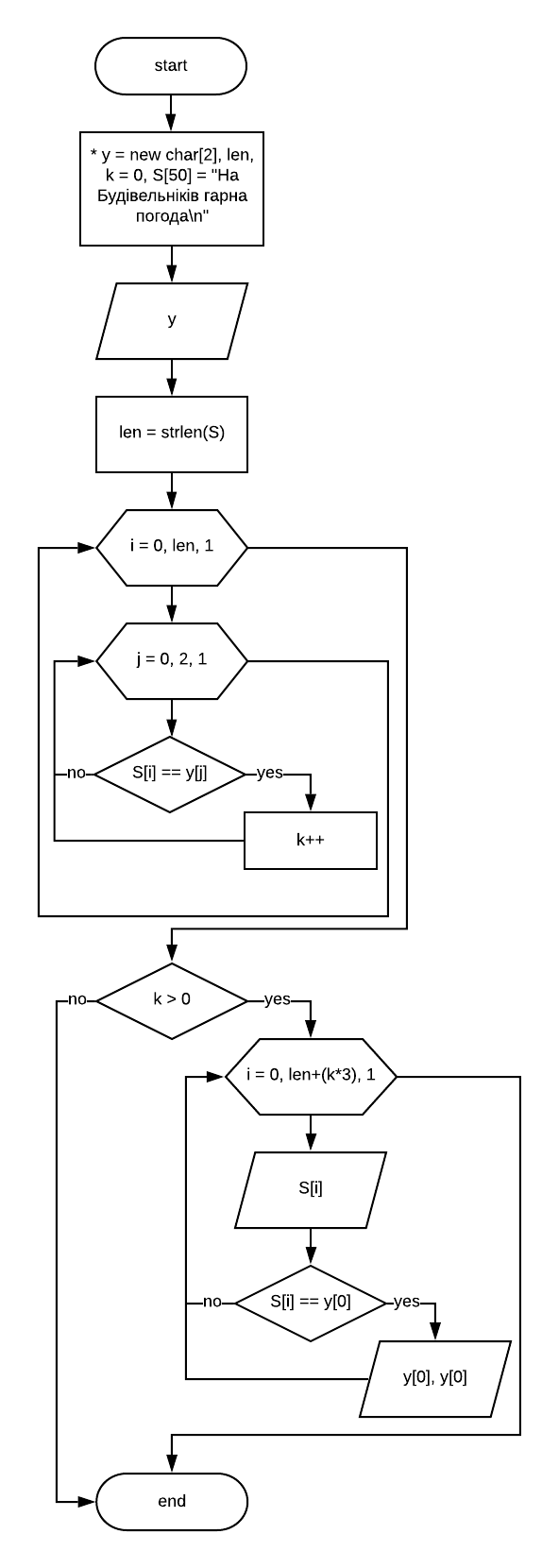
Задача 15 (символьні та рядкові величини)

Постановка задачі:

Нехай дано текст S та значення символьної змінної у. У тексті повторити тричі всі символи, що збігаються з у.

Блок-схема алгоритму:

В даній програмі дано текст S. Користувач вводить символ з клавіатури. Програма шукає цей символ у тексті. Якщо вона знаходить, виводиться на екран текст з тричі повтореним символом, що ввів користувач.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

char\* y = new char[2];

int len, k = 0;

char S[50] = "На Будівельніків гарна погода\n";

cout << S;

// ввід данних користувачем

cout << "Введіть літеру для перевірки:\n";

cin.getline(y, 2);

// обчислення

len = strlen(S);

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

for (int j = 0; j < 2; j++) {

if (S[i] == y[j]) {

k++;

}

}

}

// вивід результатів

if (k > 0) {

for (int i = 0; i < len+(k\*3); i++) {

cout << S[i];

if (S[i] == y[0]) cout << y[0] << y[0];

}

}

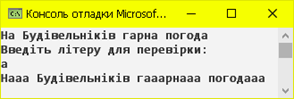
else {

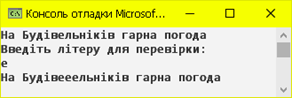
cout << "Літери не знайдено\n";

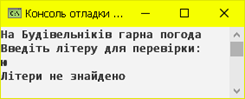
}

}

Результат виконання:







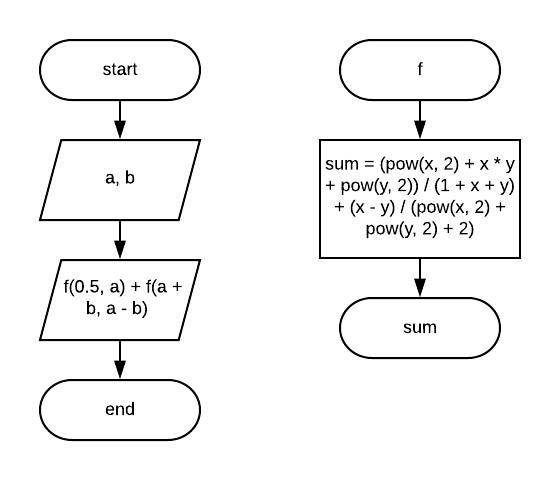
Задача 16 (функції)

Постановка задачі:

За даними дійсними числами a, b обчислити:

Блок-схема алгоритму:

В даній задачі користувач вводить значення a та b з клавіатури. Далі, функція f обчислює значення u за дано, після чого результат обчислень виводиться на екран.



Лістинг програми:

// створюємо функцію f

float f(float x, float y) {

float sum = (pow(x, 2) + x \* y + pow(y, 2)) / (1 + x + y) + (x - y) / (pow(x, 2) + pow(y, 2) + 2);

return(sum);

}

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

float a, b;

// ввід данних користувачем

cout << "Введіть: \na = "; cin >> a;

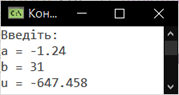
cout << "b = "; cin >> b;

// обчислення та вивід результатів

cout << "u = " << f(0.5, a) + f(a + b, a - b)<<endl;

}

Результат виконання:





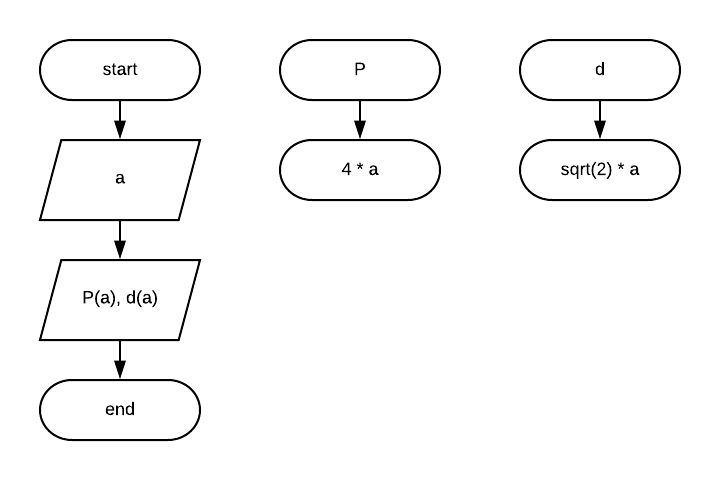
Задача 17 (функції)

Постановка задачі:

Створити підпрограму, яка б за стороною а визначала периметр квадрата та його діагональ. У програмі організувати виклик цієї підпрограми.

Блок-схема алгоритму:

В даній задачі користувач вводить значення а з клавіатури. Далі виконується виклик 2 підпрограм, що рахують периметр на діагональ квадрата з сторонами а. Результат обчислень виводиться на екран.



Лістинг програми:

// створюємо функцію P

float P(float a) {

return (4 \* a);

}

// створюємо функцію d

float d(float a) {

return (sqrt(2) \* a);

}

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251);// підключення української мови

// оголошуємо змінні

float a;

// ввід данних користувачем

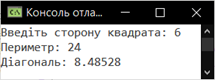
cout << "Введіть сторону квадрата: "; cin >> a;

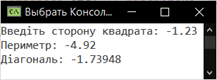
// обчислення та вивід результатів

cout << "Периметр: " << P(a) << "\nДіагональ: " << d(a) << endl;

}

Результат виконання:





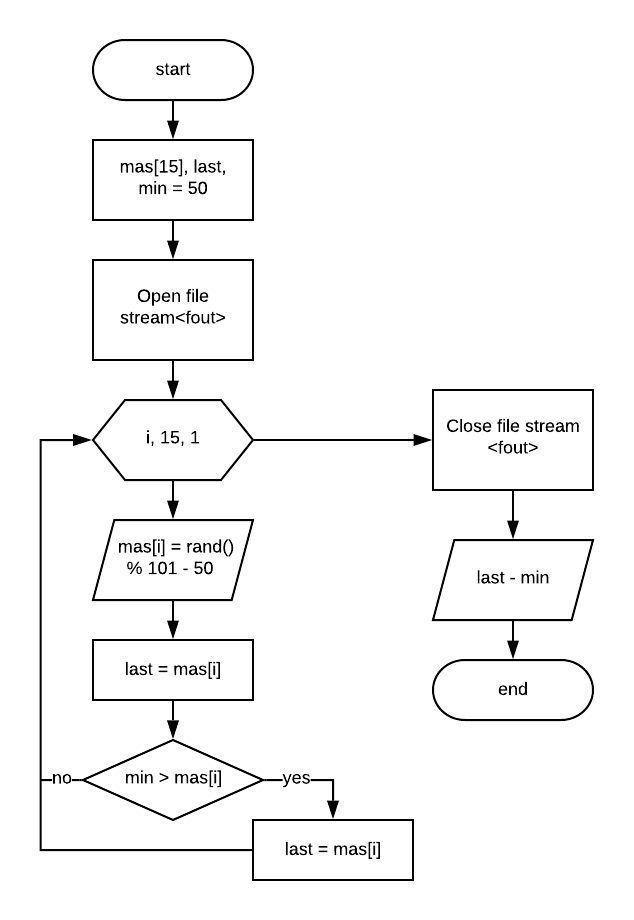
Задача 18 (файли)

Постановка задачі:

Дано файл, який містить цілі числа. Визначити різницю між останнім та найменшим елементом.

Блок-схема алгоритму:

В даній задачі масив наповнюється випадковими числами і записується у файл massiv.dat. Програма знаходить мінімальний елемент і віднімає його від останього. Після результат виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

srand(time(NULL));

// оголошуємо змінні

int mas[15], last, min = 50;

ofstream fout("massiv.dat");

if (!fout) cout << "File error\n";

// обчислення та запис результатів у файл massiv.dat

for (int i = 0; i < 15; i++)

{

mas[i] = rand() % 101 - 50;

fout << mas[i] << " ";

cout << mas[i] << " ";

last = mas[i];

if (min > mas[i]) min = mas[i];

}

fout.close();

// вивід результатів

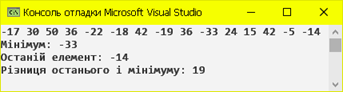
cout << "\nМінімум: " << min;

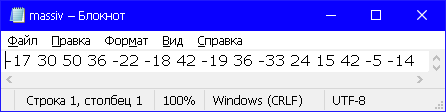
cout << "\nОстаній елемент: " << last;

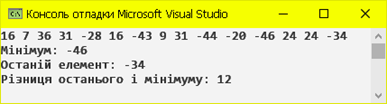
cout << "\nРізниця останього і мінімуму: " << last - min<<endl;

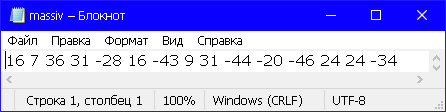
}

Результат виконання:









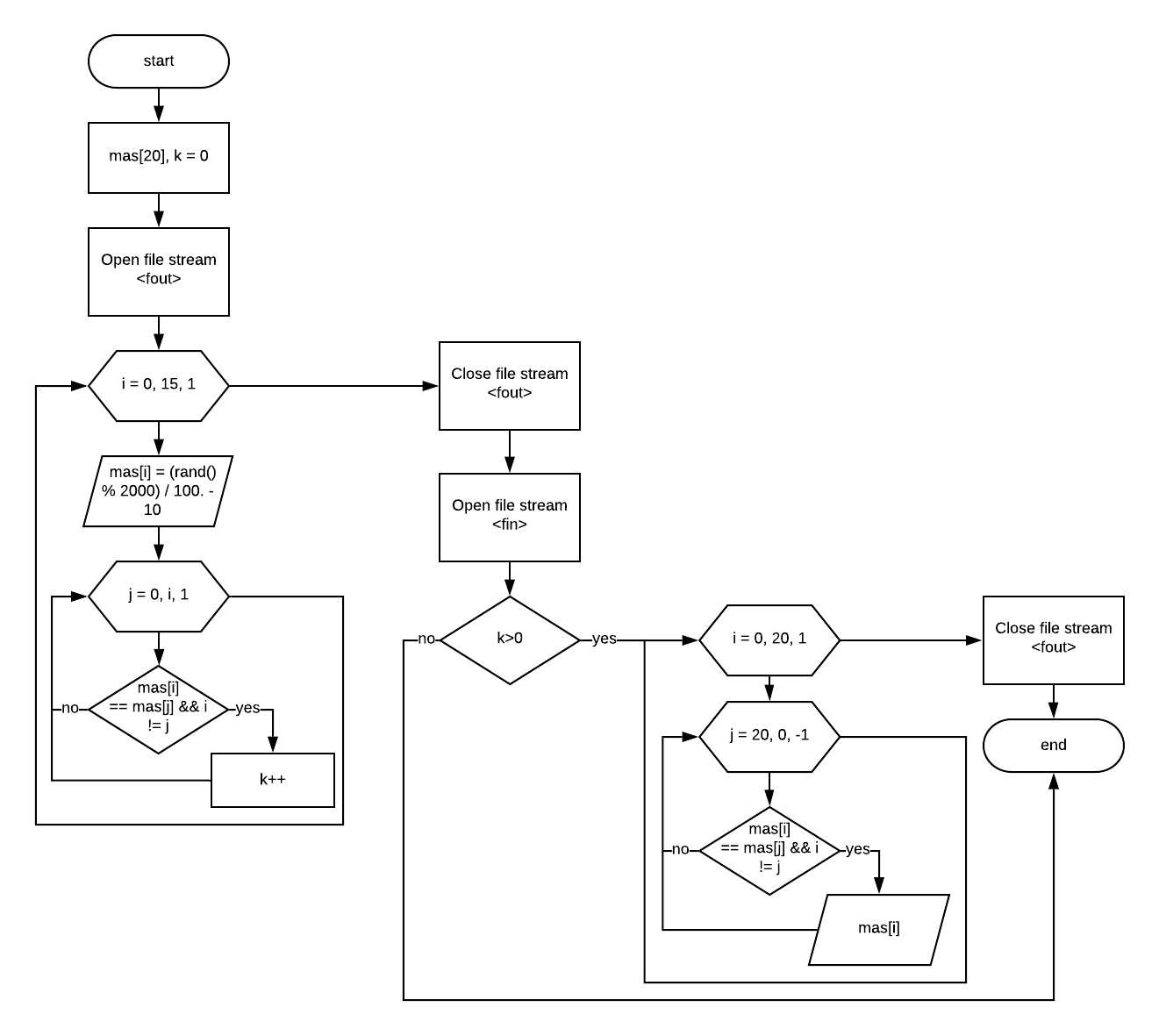
Задача 19 (файли)

Постановка задачі:

Дано файл f, що складається з дійсних елементів. Створити файл g, що містить тільки ті елементи файлу f які повторюються.

Блок-схема алгоритму:

В даній задачі масив наповнюється випадковими числами. Програма перевіряє, чи повторюються елементи у масиві. Якщо так, то у файл g записується це число. При негативному результаті файл g залишається пустим. Результат виводиться на екран.



Лістинг програми:

int main()

{

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

srand(time(NULL));

// оголошуємо змінні

float mas[20];

int k=0;

ofstream fout("f.dat");

if (!fout) cout << "File error\n";

// обчислення та запис результатів у файл massiv.dat

for (int i = 0; i < 20; i++)

{

mas[i] = (rand() % 2000) / 100. - 10;

fout << mas[i] << " ";

cout << mas[i] << " ";

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

if (mas[i] == mas[j] && i != j) k++;

}

}

fout.close();

// обчислення та вивід результатів

ofstream fin("g.dat");

if (!fin) cout << "File error\n";

if (k>0){

cout << "\nЧисла ряду, що повторюються: ";

for (int i = 0; i < 20; i++)

{

for (int j = 20; j >= i; j--){

if (mas[i] == mas[j] && i != j)

{

fin << mas[i] << " ";

cout << "\n" << mas[i] << " ";

}

}

}

}

else {

fin << "-";

cout << "\nНема однакових елементів у ряді\n";

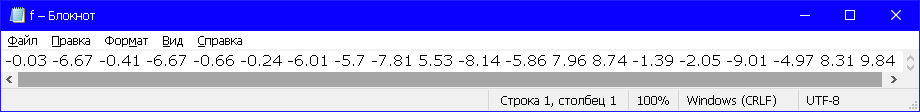
}

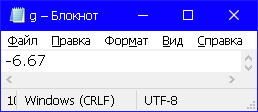
fin.close();

}

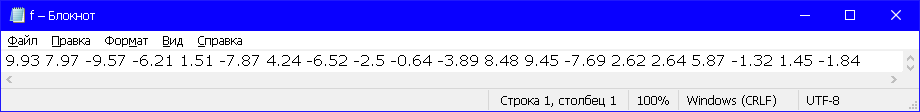
Результат виконання:

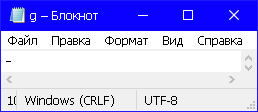












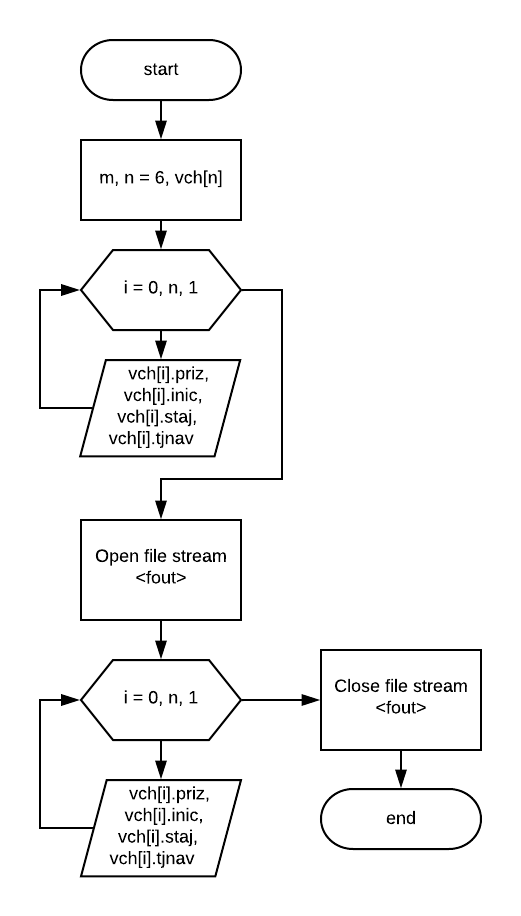
Задача 20 (структури)

Постановка задачі:

Роздрукувати список вчителів школи, які викладають математику і інформатику з зазначенням стажу їх роботи і тижневого навантаження.

Блок-схема алгоритму:

В даній задачі користувач вводить призвіща, ініціали, стаж та тижневе навантаження з клавіатури. Після чого ці дані записуються у файл та виводяться на екран.



Лістинг програми:

// оголошення структури

struct Vchiteli {

char priz[10];

char inic[4];

int staj;

float tjnav;

};

int main() {

SetConsoleCP(1251);

SetConsoleOutputCP(1251); // підключення української мови

// оголошуємо змінні

float m;

const int n = 6;

Vchiteli vch[n];

cout << "Список вчителів, що викладають математику і інформатику\n";

ofstream fout("vch.dat");

if (!fout) cout << "Error";

// ввід данних користувачем

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

{

cout << "Призвіще: "; cin >> vch[i].priz;

cout << "Ініціали: "; cin >> vch[i].inic;

cout << "Стаж роботи: "; cin >> vch[i].staj;

cout << "Тижневе навантаження: "; cin >> vch[i].tjnav; cout << endl;

}

// вивід результатів

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

{

fout << setw(10) <<vch[i].priz << "\t" << vch[i].inic << "\t" << vch[i].staj << "\t" << vch[i].tjnav << "\t" << endl;

cout << setw(10) <<vch[i].priz << "\t" << vch[i].inic << "\t" << vch[i].staj << "\t" << vch[i].tjnav << "\t" << endl;

}

fout.close();

}

Результат виконання:

