

**Міністерство освіти і науки України
Національний університет «Львівська політехніка»
Кафедра систем штучного інтелекту**



Звіт

**про виконання розрахунково-графічних робіт блоку № 7
з дисципліни: «Основи програмування»**

до:

**ВНС Розрахунково-графічних робіт № 1-4
Практичних Робіт до блоку № 7**

Виконала:

**Студентка групи ШІ-12
Смачило Іванна Ростиславівна**

Львів 2024

VNS practice work 1 task №17

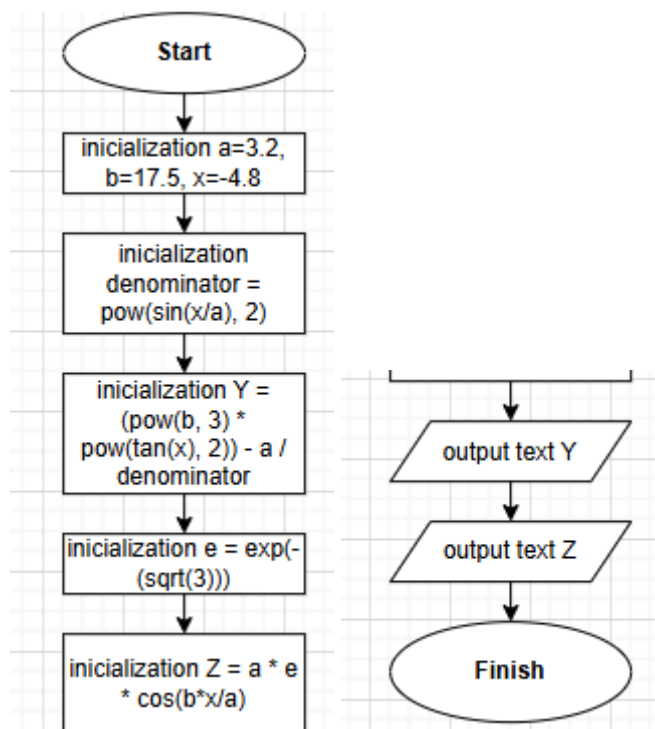
В коді використані дійсні змінні, використані дійсні змінні з подвійною точністю, математичні операції та математичні функції, оператор виведення, структури даних.

```
1  #include <iostream>
2  #include <cmath>
3  #include <stdio.h>
4
5  using namespace std;
6
7  int main()
8  {
9      float a = 3.2, b = 17.5, x = -4.8; //надала значення a,b,x через float, бо числа з плаваючою крапкою
10
11     double denominator = pow(sin(x/a), 2); //використала double(для чисел з плаваючою крапкою), pow-для степеня
12     double Y = (pow(b, 3) * pow(tan(x), 2)) - a / denominator;
13
14     double e = exp(-(sqrt(3))); //e=exp, корінь=sqrt
15     double Z = a * e * cos(b*x/a);
16
17     cout << "Result Y = " << Y << "\n";
18     cout << "Result Z = " << Z;
19
20     return 0;
21 }
```

Результат:

```
Result Y = 694651
Result Z = 0.248057
```

Блок-схема:



VNS practice work 2 task №14

В кодi використано дійсні змінні, змінні з подвійною точністю, умовні оператори та розгалуження, одновимірний масив, for цикл, математичні операції та математичні функції, оператор введення, структури даних, вказівники, do while цикл.

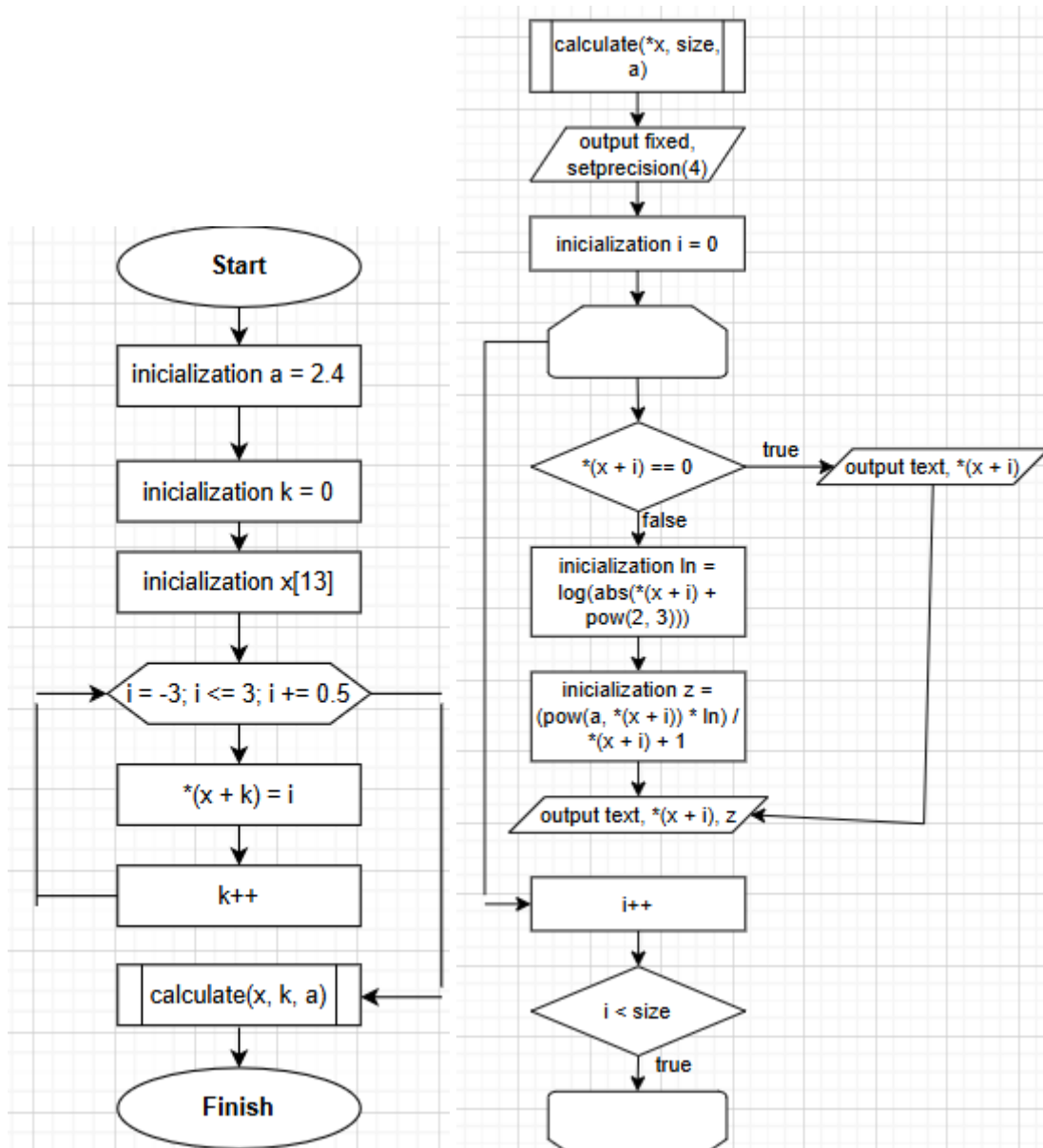
```
1  #include <iostream>
2  #include <cmath>
3  #include <stdio.h>
4  #include <iomanip>
5
6  using namespace std;
7
8  void calculate(double *x, int size, double a)
9  {
10     cout << fixed << setprecision(4);
11     int i = 0;
12     do
13     {
14         if(*(x + i) == 0)
15         {
16             cout << "z is undefined for x = " << *(x + i) << " (division by 0)" << "\n";
17         }
18         else
19         {
20             double ln = log(abs(*(x + i) + pow(2, 3)));
21             double z = (pow(a, *(x + i)) * ln) / *(x + i) + 1;
22             cout << "x for x: " << *(x + i) << " is " << z << "\n";
23         }
24         i++;
25     }
26     while (i < size);
27 }
```

```
26
27 int main()
28 {
29     double a = 2.4;
30     int k = 0;
31     double x[13]; //значення від -3 до 3
32     for(double i = -3; i <= 3; i += 0.5)
33     {
34         *(x + k) = i;
35         k++;
36     }
37
38     calculate(x, k, a);
39
40     return 0;
41 }
```

Результат:

```
x for x: -3.0000 is 0.9612
x for x: -2.5000 is 0.9236
x for x: -2.0000 is 0.8445
x for x: -1.5000 is 0.6644
x for x: -1.0000 is 0.1892
x for x: -0.5000 is -1.6012
z is undefined for x = 0.0000 (division by 0)
x for x: 0.5000 is 7.6308
x for x: 1.0000 is 6.2733
x for x: 1.5000 is 6.5803
x for x: 2.0000 is 7.6314
x for x: 2.5000 is 9.3929
x for x: 3.0000 is 12.0495
```

Блок-схема:



VNS practice work 3 task №9

В кодї використані цілочисельні змінні, дійсні змінні, дійсна змінна з подвійною точністю, умовні оператори та розгалуження, функції роботи з файлами, для того, щоб записати у файл і щоб зчитати з файлу, оператори виведення і введення даних, структури даних, while цикл.

```

1  #include <iostream>
2  #include <iomanip>
3  #include <fstream>
4
5  using namespace std;
6
7  int main()
8  {
9
10     double r1, r2, r;
11
12     cout << "Обчислення опору електричного ланцюга при паралельному з'єднанні елементів:" << "\n";
13     cout << "Введіть початкові дані:" << "\n";
14     cout << "Величина першого опору (Ом) > ";
15     cin >> r1;
16     cout << "Величина другого опору (Ом) > ";
17     cin >> r2;
18
19     ofstream file("result.txt");
20     while(!file.is_open())
21     {
22         cerr << "Помилка відкриття файлу!" << "\n";
23         return 1;
24     }
25
26     //розрахунок загального опору
27     if (r1 > 0 && r2 > 0)
28     {
29         r = r1 + r2;
30
31         file << fixed << setprecision(2);
32         file << "Обчислення опору електричного ланцюга при паралельному з'єднанні елементів:" << "\n";
33         file << "Величина першого опору (Ом): " << r1 << "\n";
34         file << "Величина другого опору (Ом): " << r2 << "\n";
35         file << "Опір ланцюга: " << r << " Ом" << "\n";
36     }
37
38     int pagesTotal = 948;
39     int words = 30;
40     int letters = 5;
41     double width = 2.0;
42
43     int totalWords = pagesTotal * words;
44     int totalLetters = totalWords * letters;
45     double totalLength = totalLetters * width;
46
47     cout << "Кількість російських слів  $\varnothing$  словнику: " << totalWords << "\n";
48     cout << "Довжина відрізка російських слів(мм): " << totalLength << " мм" << "\n";
49     cout << "Довжина відрізка російських слів(м): " << totalLength / 1000 << " м" << "\n";
50
51     file << "Кількість російських слів  $\varnothing$  словнику: " << totalWords << "\n";
52     file << "Довжина відрізка російських слів(мм): " << totalLength << " мм" << "\n";
53     file << "Довжина відрізка російських слів(м): " << totalLength / 1000 << " м" << "\n";
54
55     file.close();
56
57     cout << "Ваші результати було записано  $\varnothing$  файл result.txt" << "\n";
58
59     return 0;
60 }

```

Результат:

Введіть початкові дані:

Величина першого опору (Ом) > 15

Величина другого опору (Ом) > 27.3


Опір ланцюга: 42.30 Ом

Кількість російських слів у словнику: 28440

Довжина відрізка російських слів(мм): 284400.00 мм

Довжина відрізка російських слів(м): 284.40 м

Ваші результати було записано у файл result.txt

C: > Epic_7 >  result.txt

1 Обчислення опору електричного ланцюга при паралельному з'єднанні елементів:

2 Величина першого опору (Ом): 15.00

3 Величина другого опору (Ом): 27.30

4 Опір ланцюга: 42.30 Ом

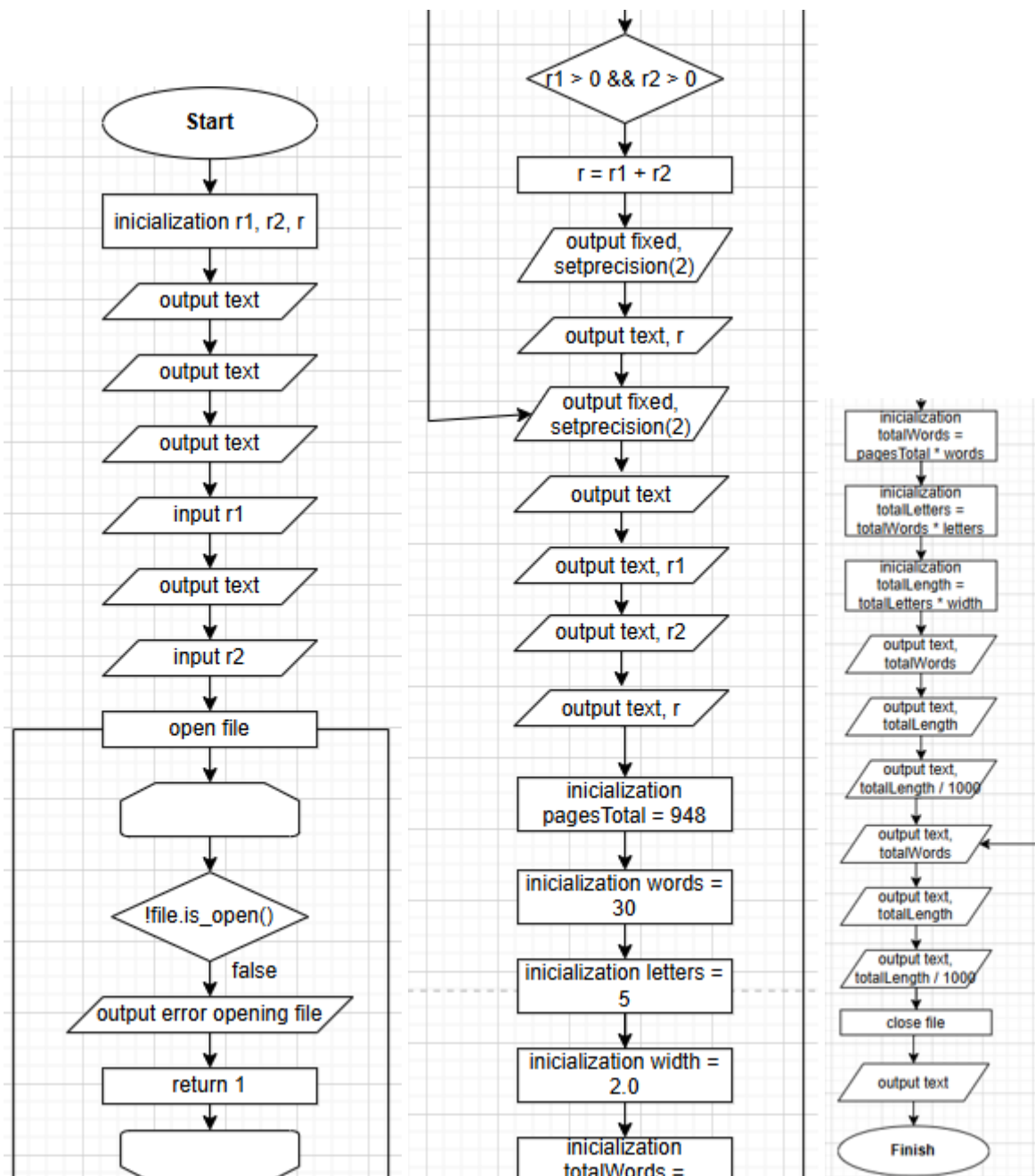
5 Кількість російських слів у словнику: 28440

6 Довжина відрізка російських слів(мм): 284400.00 мм

7 Довжина відрізка російських слів(м): 284.40 м

8

Блок-схема:



VNS practice work 4 task №12

В коді використано цілочисельну змінну, дійсна змінна, цілочисельна константа, умовні оператори та розгалуження, for цикл, параметри та аргументи функцій, оператори виведення та введення даних, структура даних, оператори break і continue.

```

1  #include <iostream>
2
3  using namespace std;
4
5  //рекурсивна функція для обчислення суми елементів масиву
6  double calculate(double arr[], int ind, int size)
7  {
8      double sum = 0;
9      for(int i = 0; i < size; i++)
10     {
11         if(arr[i] < 0)
12         {
13             continue;
14         }
15         sum += arr[i];
16         if(sum > 100)
17         {
18             break;
19         }
20     }
21     return sum;
22 }
23

```

```

24 int main()
25 {
26     const int size = 5; //цілочисельна константа
27     double numbers[size];
28     double sum = 0;
29
30     cout << "Enter " << size << " fractional numbers: " << "\n";
31     for(int i = 0; i < size; i++)
32     {
33         cout << "Number " << i + 1 << " : ";
34         cin >> numbers[i];
35     }
36
37     sum = calculate(numbers, 0, size);
38     double average = sum / size;
39     cout << "Arithmetic mean for this numbers is: " << average << "\n";
40
41     return 0;
42 }

```

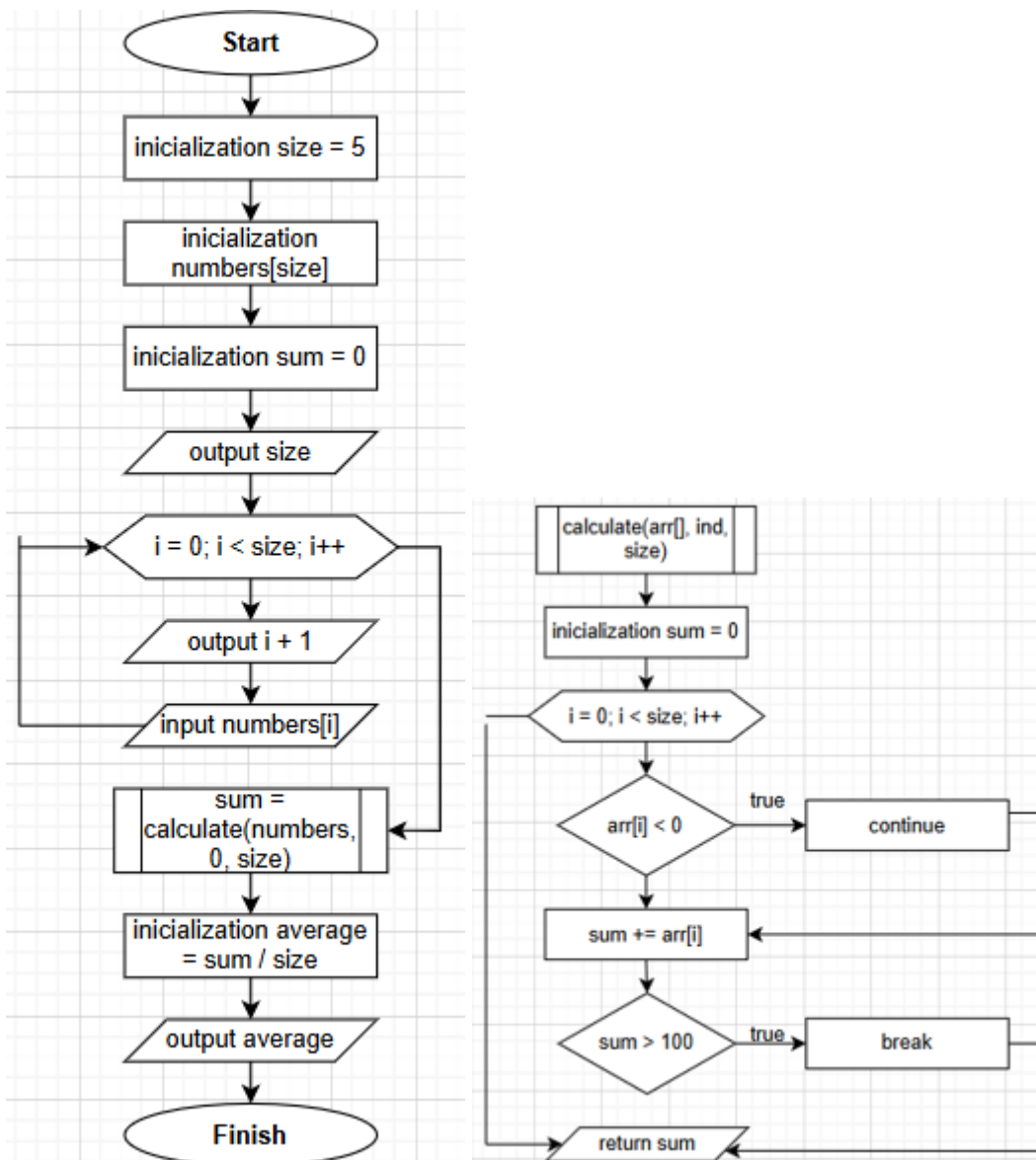
Результат:

```

Enter 5 fractional numbers:
Number 1 : 1.6
Number 2 : 4.7
Number 3 : 2.9
Number 4 : 7.4
Number 5 : 9.4
Arithmetic mean for this numbers is: 5.2

```


Блок-схема:



Algotester problem “Непарний масив”

В кодї використано умовні оператори та розгалуження, for цикл, параметри та аргументи функції, математичні операції та математичні функції, перевантаження функції, оператори виведення та введення даних.

```

1 //Непарний масив
2 #include <iostream>
3 #include <vector>
4 #include <cmath>
5
6 using namespace std;
7
8 //в коді використано параметри та аргументи функції, математичні операції та математичні функції, свою структуру дан
9 void numbers(const vector<int>& arr, vector<int>& even, vector<int>& odd)
10 {
11     for(int num : arr)
12     {
13         if(num % 2 == 0)
14         {
15             even.push_back(num);
16         }
17         else
18         {
19             odd.push_back(num);
20         }
21     }
22 }

```

```

23
24 void vectors(const vector<int>& large, vector<int>& small, vector<int>& result)
25 {
26     for(int i = 0; i < large.size(); i++)
27     {
28         result.push_back(large[i]);
29         if(i < small.size())
30         {
31             result.push_back(small[i]);
32         }
33     }
34 }
35
36 void vectors(const vector<double>& large, const vector<double>& small, const vector<double>& result)
37 {
38     for(int i = 0; i < large.size(); i++)
39     {
40         result.push_back(large[i]);    no instance of overloaded function "std::vector<_Tp, _Alloc>::push_back [with
41         if(i < small.size())
42         {
43             result.push_back(small[i]);    no instance of overloaded function "std::vector<_Tp, _Alloc>::push_back [
44         }
45     }
46 }
47
48 int main()
49 {
50     int n;
51     cin >> n;
52
53     for(int i = 0; i < n; i++)
54     {
55         cin >> arr[i];    identifier "arr" is undefined
56     }
57
58     numbers(arr, even, odd);    identifier "arr" is undefined

```

```

60     if(abs((int)even.size() - (int)odd.size()) > 1)
61     {
62         cout << -1 << "\n";
63         return 0;
64     }
65
66     vector<int> f;
67
68     if(even.size() >= odd.size())
69     {
70         vectors(even, odd, f);
71     }
72     else
73     {
74         vectors(odd, even, f);
75     }
76
77     for(int num : f)
78     {
79         cout << num << " ";
80         cout << "\n";
81     }
82
83     return 0;
84 }

```

Результат:

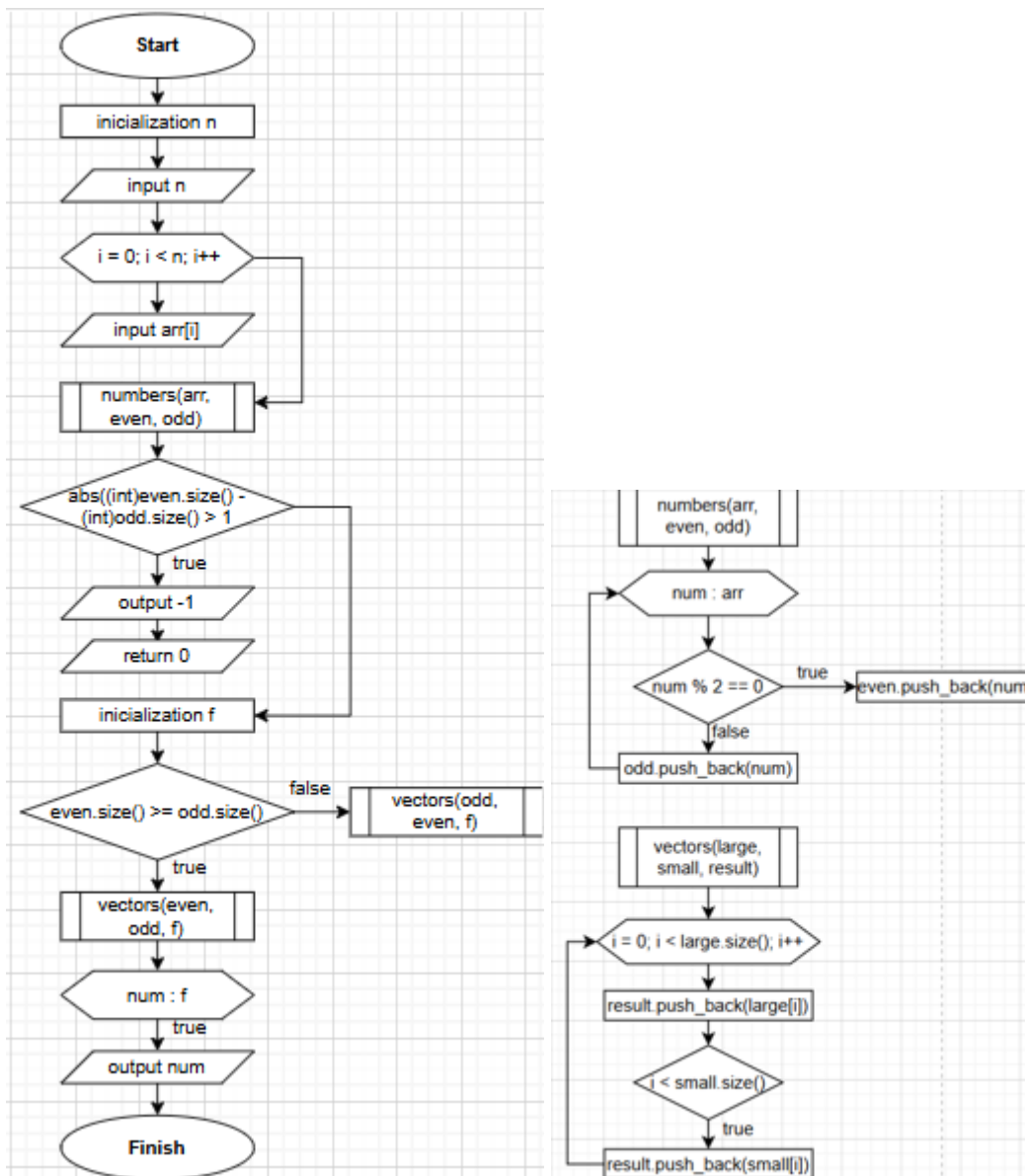
```

3
4 7 47
7
4
47

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.063	2.367	Перегляд

Блок-схема:



Algotester problem “Загадкове число”

В кодї використано цілочисельні змінні, умовні оператори та розгалуження, одновимірний масив, for цикл, математичні операції та математичні функції, оператори виведення та введення даних, структури даних.

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main()
7  {
8      string number;
9      cin >> number;
10
11     int minNum = 0, maxNum = 0;
12     int multip = 1;
13
14     for (int i = number.size() - 1; i >= 0; i--)
15     {
16         if (number[i] == '*')
17         {
18             minNum += (i == 0) ? 1 * multip : 0;
19             maxNum += 9 * multip;
20         }
21         else
22         {
23             minNum += (number[i] - '0') * multip;
24             maxNum += (number[i] - '0') * multip;
25         }
26         multip *= 10;
27     }
28
29     cout << minNum << " " << maxNum << "\n";
30
31     return 0;
32 }

```

Результат:

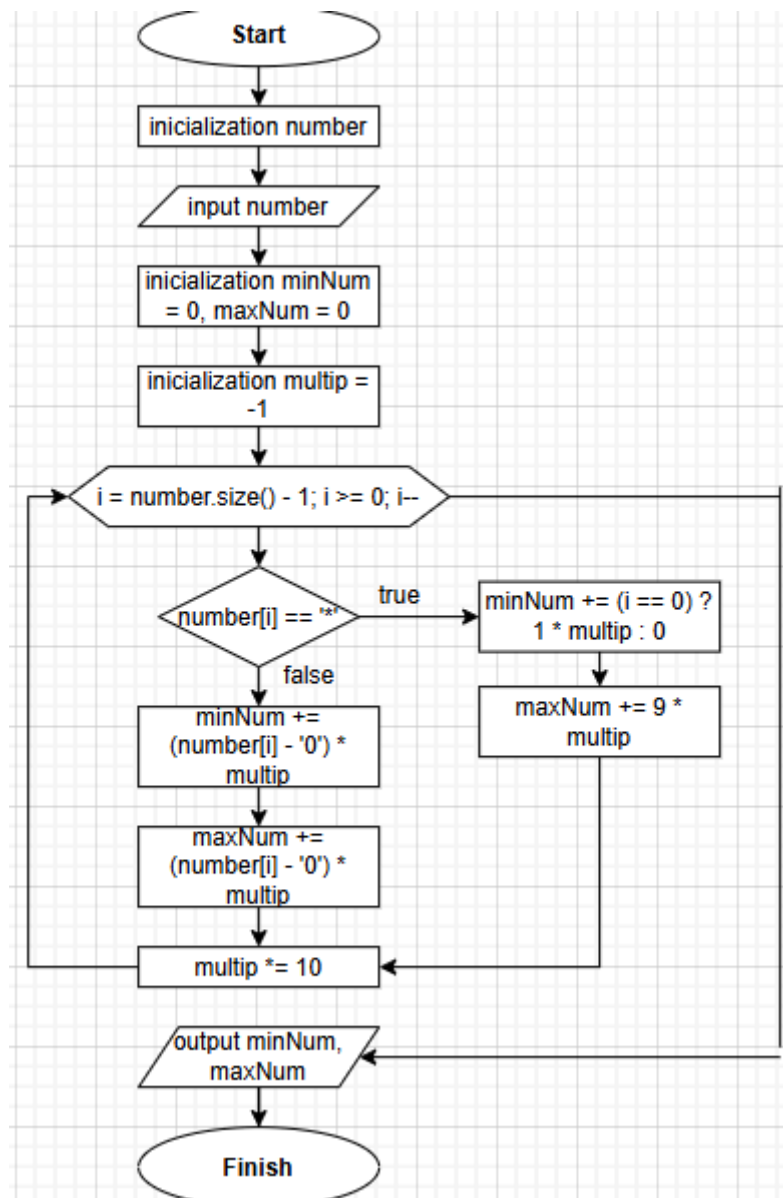
```

47
47 47

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
хвилину тому	C++ 23	Зараховано	0.002	1.117	Перегляд

Блок-схема:



Algotester problem “Добра справа”

В кодї використано математичні операції та математичні функції, оператори виведення та введення даних.

```

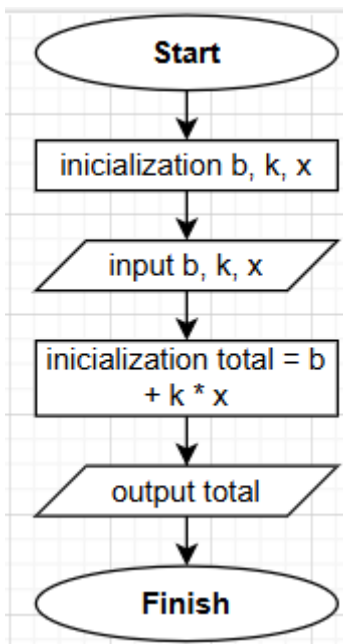
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int b, k, x;
8      cin >> b >> k >> x;
9
10     int total = b + k * x;
11
12     cout << total << "\n";
13
14     return 0;
15 }
  
```

Результат:

47 4 7
75

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.002	1.039	Перегляд

Блок-схема:



Algotester problem “Вогняне дихання”

В коді використано умовні оператори та розгалуження, одновимірний масив, for цикл, параметри та аргументи функції, математичні операції та математичні функції, оператори виведення та введення даних.

```

1  #include <iostream>
2  #include <algorithm>
3  #include <iomanip>
4  #include <cmath>
5
6  using namespace std;
7
8  struct coordinate
9  {
10     int X, Y;
11 };
12
13 double distance(int X1, int Y1, int X2, int Y2)
14 {
15     return sqrt(static_cast<double>(pow(X1 - X2, 2) + pow(Y1 - Y2, 2)));
16 }
17
18 int main()
19 {
20     coordinate r;
21     cin >> r.X >> r.Y;
22
23     int n, k;
24     cin >> n >> k;
25
26     coordinate point[1000];
27     double Distance[1000];
28
29     for(int i = 0; i < n; i++)
30     {
31         cin >> point[i].X >> point[i].Y;
32     }

```

```

33
34     for(int i = 0; i < n; i++)
35     {
36         Distance[i] = distance(r.X, r.Y, point[i].X, point[i].Y);
37     }
38
39     sort(Distance, Distance + n);
40     cout << setprecision(13) << Distance[k - 1];
41     return 0;
42 }

```

Результат:

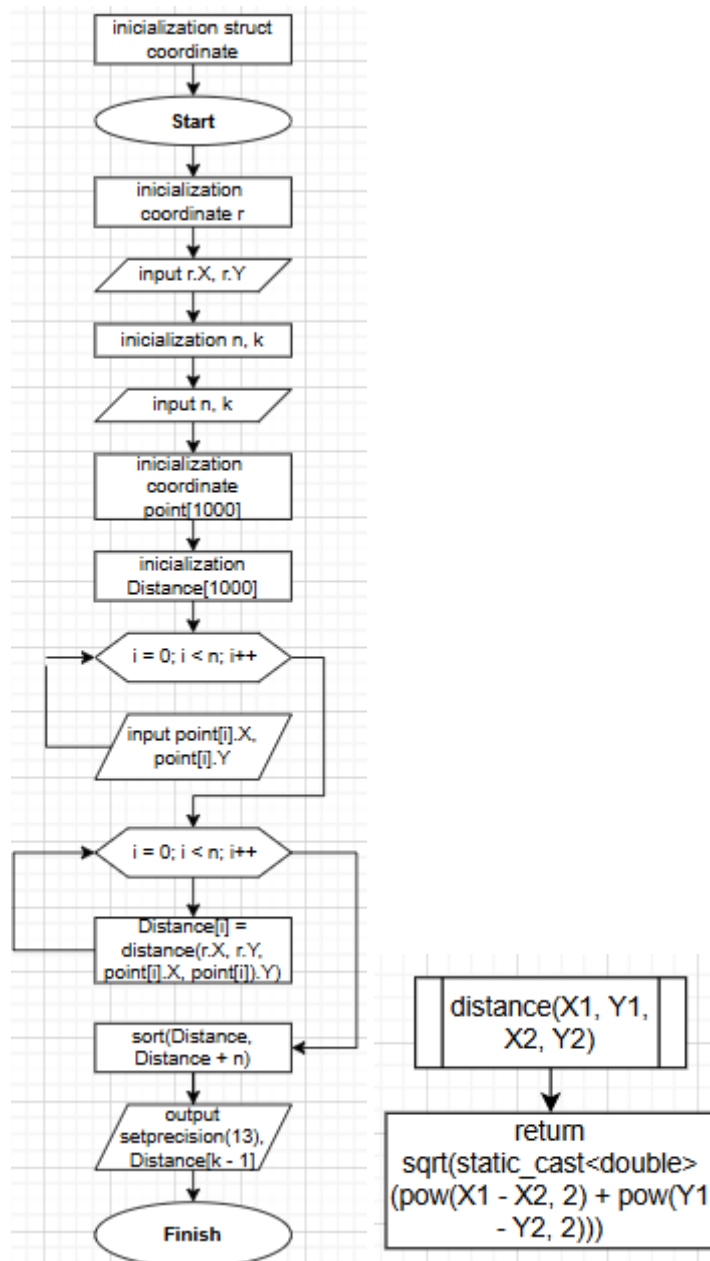
```

5 6
4 2
3 4
8 11
6 5
17 25
2.828427124746

```

Створено	Компілятор	Результат	Час (сек.)	Пам'ять (МіБ)	Дії
декілька секунд тому	C++ 23	Зараховано	0.003	1.176	Перегляд

Блок-схема:



Висновок: в ході роботи над даною розрахунковою роботою(сагою) я набула практичних навичок в розв'язанні задач різної складності, використовуючи перелічені умови: використання цілочисельних змінних, дійсні змінні, дійсні з подвійною точністю змінні, цілочисельні константи, умовні оператори та розгалуження, масиви, while цикл, for цикл, оператори break і continue, параметри та аргументи функції, математичні операції та математичні функції, оператори виведення та введення даних, свої структури даних. Також розвинула свої вміння в побудові блок-схем різної складності.