

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-11

Андрусишин Соломія Володимирівна

**Мета роботи:** одержати практичні навички в розробці і дослідженні алгоритмів розв'язання задач.

## **Виконання роботи:**

1) Опрацювання завдання та вимог до програми та середовища

### **Завдання №1 VNS Practice Work - Task 1**

**Варіант 4.** Площу поверхні зрізаного конуса і його об'єм за формулами:  $S = \pi(R+r)l + \pi R^2 + \pi r^2$ ;  $l = \sqrt{h^2 + (R-r)^2}$ ;

$V = \frac{\pi}{3}(R^2 + r^2 + Rr)h$ . Значення  $R$ ,  $r$ ,  $h$  вибрати самостійно.

### **Завдання №2 VNS Practice Work - Task 2**

**Варіант 15.**  $z = \frac{\sqrt{x-1,5} + x^a}{(x-2)^{1/3}}$ ;  $x \in [1,4]$ ;  $h_x = 0,5$ ;  $a > -0,5$ ;

$h_a = 0,2$ , де  $x$  і  $a$  змінюються одночасно.

### **Завдання №3 VNS Practice Work - Task 3**

**Варіант 23.** Обчислення об'єму куба. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму куба.

Введіть довжину ребра (см) і натисніть клавішу <Enter> **9.5**

Об'єм куба: 857.38 куб.см.

Обчислення опору електричного ланцюга, що складається з двох паралельно сполучених резисторів.

### **Завдання №4 VNS Practice Work - Task 4**

**Варіант 12.** Написати програму, яка вводиться з клавіатури 5 дробових чисел і обчислює їх середнє арифметичне.

## Завдання №5 Algotester

### Марічка і печиво

Зібралися Зеник і Марічка разом з пластунами в похід. Похід — серйозна справа. Потрібно запаситись продуктами харчування та розподілити їх споживання по днях так, щоб всім вистачило. Цього разу Зеник слідкує за тим, щоб печива вистачило аж до останнього дня походу. Зеник чітко знає, скільки пачок печива повинно залишитись кожного дня, і щовечора перераховує їх. Якщо Зеник побачить, що залишилось менше пачок, ніж повинно залишитись за його розрахунками, він неодмінно знайде того, хто з'їв забагато печива, і покарає його.

Марічка дуже любить печиво. Сьогодні, коли всі пластуни покинуть свої намети і підуть купатися в річку, Марічка планує непомітно з'їсти трохи печива. Звісно, Марічка не хоче бути покараною і дуже боїться, щоб Зеник не помітив пропажу.

Марічка підгледіла, скільки пачок печива є в рюкзаку Зеника. Також вона знає, скільки штук в кожній пачці. Марічці не терпиться дізнатися, скільки ж печива вона зможе з'їсти так, щоб Зеник не помітив. Зеник помітить пропажу печива з деякої пачки тоді і тільки тоді, коли Марічка повністю спустошить її.

### Вхідні дані

У першому рядку задано одне натуральне число  $n$  — кількість пачок печива.  
У другому рядку задано  $n$  натуральних чисел  $a_i$  — кількість штук печива в  $i$ -й пачці.

### Вихідні дані

У єдиному рядку виведіть одне ціле число — максимальну кількість штук печива, яку зможе з'їсти Марічка так, щоб Зеник не помітив цього.

### Обмеження

20% тестів:

$$1 \leq n \leq 1000, 0 \leq a_i \leq 10^4$$

60% тестів:

$$1 \leq n \leq 10^5, 0 \leq a_i \leq 10^4$$

20% тестів:

$$1 \leq n \leq 10^5, 0 \leq a_i \leq 10^9$$

## Завдання №6 Algotester

# Торт для Петрика

Обмеження: 2 сек., 256 MiB

Зовсім скоро в Петрика день народження, і він з нетерпінням чекає цього свята. Його друзі слоненята вирішили зробити йому подарунок і приготувати ідеальний торт.

На думку слоненят, ідеальний торт має складатися з рівно  $n$  ярусів, кожен з яких є циліндром. Проте при визначенні ідеальності висоти циліндрів не важливі, важливі лише їхні радіуси, тому яруси можа вважати кругами. Яруси кладуть один на одного від найбільшого до найменшого так, щоб вони мали спільний центр. Оскільки всі яруси є однаково гарними, і важливістю жодного не можна нехтувати, торт вважається ідеальним тільки тоді, коли видимі площі всіх  $n$  ярусів рівні, якщо дивитися на торт зверху.

Слоненята вирішили, що радіус найбільшого ярусу їхнього торта має бути  $r$ . Допоможіть слоненятам обчислити, яким же буде радіус найменшого.

## Вхідні дані

У єдиному рядку задано два цілих числа  $n$  та  $r$  — відповідно кількість ярусів та радіус найбільшого ярусу в ідеальному торті.

## Вихідні дані

У єдиному рядку виведіть дійсне число — радіус найменшого ярусу. Відповідь уважатиметься правильною, якщо її абсолютна чи відносна похибка не перевищуватиме  $10^{-4}$ .

## Обмеження

$$1 \leq n \leq 100, 1 \leq r \leq 100,$$

$$1 \leq r \leq 100, 1 \leq r \leq 100.$$

## Приклади

## Завдання №7 Algotester

# Офісна Вулиця. Частина 1

Обмеження: 2 сек., 256 MiB

Зустрілися якось працівники великих компаній і почали... Обговорювати план вулиці.

Виявляється, всі приміщення, які орендуватимуть ці компанії, збудують вздовж однієї вулиці.

$i$ -та компанія орендуватиме офіс довжиною  $l_i$  метрів. Офіси будуватимуть один за одним, починаючи з точки 0. Всі працівники приїжджатимуть на стоянку, яку побудують в точці 0, та будуть йти до офісів своїх компаній. Тобто, якщо офіси будуть збудовані в порядку  $p_1, p_2, \dots, p_n$ , то перший офіс почнеться в точці 0 і закінчиться в точці  $l_{p_1}$ , другий почнеться в  $l_{p_1}$  і закінчиться в  $l_{p_1} + l_{p_2}$  і т.д. Двері кожного офісу завжди є в кінці будинку, який є ближчим до стоянки.

Ваше завдання — допомогти розмістити офіси компаній на цій вулиці в такому порядку, щоб сумарна відстань від точки 0 до усіх офісів була мінімальною.

## Вхідні дані

У першому рядку задане ціле число  $n$  — кількість компаній.

У наступному рядку задано  $n$  цілих чисел  $l_i$  через пробіл — довжини офісів усіх компаній.

## Вихідні дані

У єдиному рядку виведіть  $n$  чисел від 1 до  $n$  — порядок компаній, в якому варто будувати офіси.

Якщо існує декілька оптимальних порядків — виведіть будь-який із них.

## Обмеження

$$1 \leq n \leq 10^5, 1 \leq l_i \leq 10^5,$$

$$1 \leq l_i \leq 10^4, 1 \leq l_i \leq 10^4.$$

## Завдання №8 Algotester

# Найбільша зростаюча підпоследовність

Обмеження: 2 сек., 256 MiB

Вам задано последовність із  $n$  цілих чисел  $a_i$ . Ваша задача — знайти довжину найбільшої зростаючої підпоследовності заданої последовності.

## Вхідні дані

У першому рядку задано ціле число  $n$ .

У наступному рядку задано  $n$  цілих чисел — последовність  $a_i$ .

## Вихідні дані

Виведіть довжину найбільшої зростаючої підпоследовності.

## Обмеження

$$1 \leq n \leq 1001 \leq n \leq 100,$$

$$1 \leq a_i \leq 10^9 \quad 1 \leq a_i \leq 10^9.$$

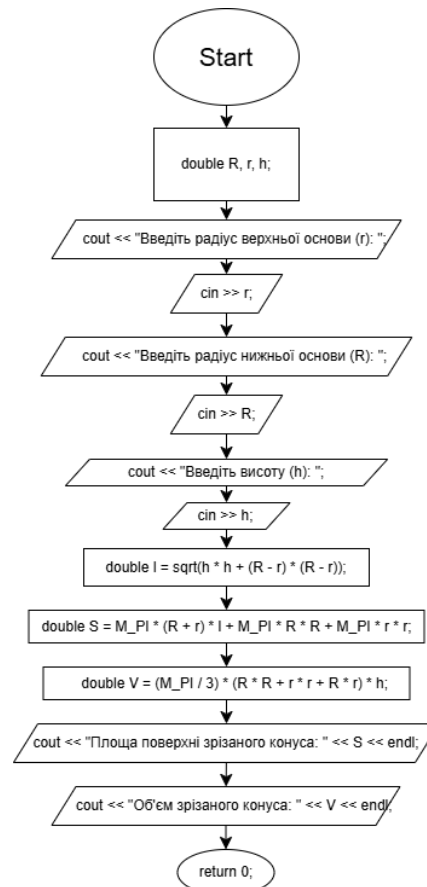
## *Виконання роботи:*

Завдання №1 VNS Practice Work - Task 1

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double R, r, h;
8      cout << "Введіть радіус верхньої основи (r): ";
9      cin >> r;
10     cout << "Введіть радіус нижньої основи (R): ";
11     cin >> R;
12     cout << "Введіть висоту (h): ";
13     cin >> h;
14
15     double l = sqrt(h * h + (R - r) * (R - r));
16
17     double S = M_PI * (R + r) * l + M_PI * R * R + M_PI * r * r;
18
19     double V = (M_PI / 3) * (R * R + r * r + R * r) * h;
20
21     cout << "Площа поверхні зрізаного конуса: " << S << endl;
22     cout << "Об'єм зрізаного конуса: " << V << endl;
23
24     return 0;
25 }
26

```

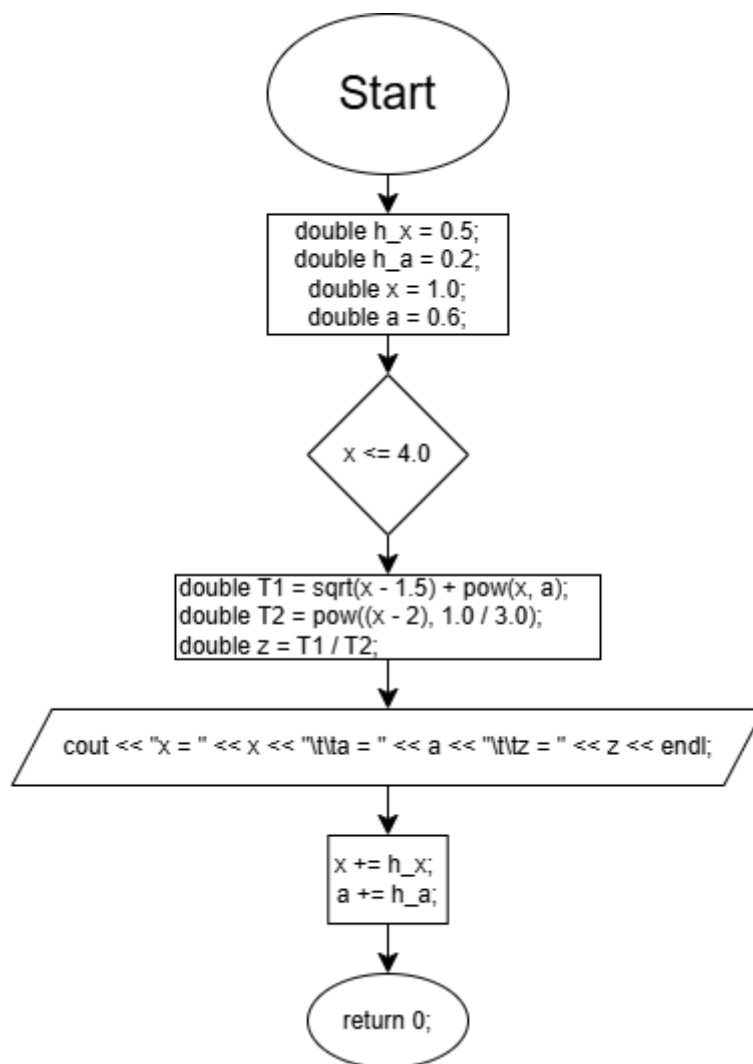


Введіть радіус верхньої основи (r): 8  
Введіть радіус нижньої основи (R): 8  
Введіть висоту (h): 9  
Площа поверхні зрізаного конуса: 854.513  
Об'єм зрізаного конуса: 1809.56

## Завдання №2 VNS Practice Work - Task 2

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main()
7  {
8      double h_x = 0.5;
9      double h_a = 0.2;
10     double x = 1.0;
11     double a = 0.6;
12
13     while (x <= 4.0)
14     {
15
16         double T1 = sqrt(x - 1.5) + pow(x, a);
17         double T2 = pow((x - 2), 1.0 / 3.0);
18         double z = T1 / T2;
19
20         cout << "x = " << x << "\t\t a = " << a << "\t\t z = " << z << endl;
21
22         x += h_x;
23         a += h_a;
24     }
25
26     return 0;
27 }
28
```

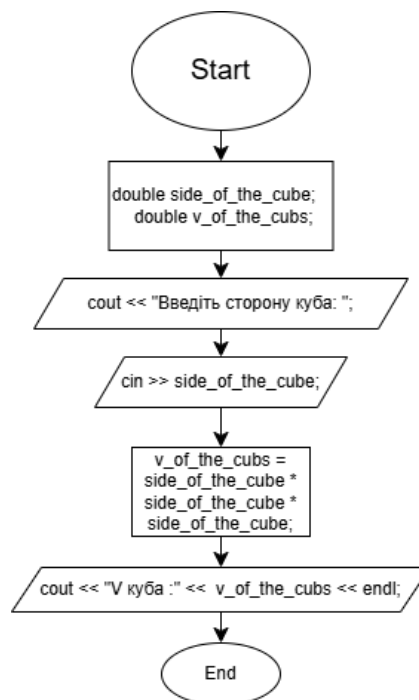




x = 1	a = 0.6	z = nan
x = 1.5	a = 0.8	z = nan
x = 2	a = 1	z = inf
x = 2.5	a = 1.2	z = 5.04323
x = 3	a = 1.4	z = 5.88028
x = 3.5	a = 1.6	z = 7.71896
x = 4	a = 1.8	z = 10.8792

### Завдання №3 VNS Practice Work - Task 3

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      double side_of_the_cube;
7      double v_of_the_cubs;
8
9      cout << "Введіть сторону куба: ";
10     cin >> side_of_the_cube;
11
12     v_of_the_cubs = side_of_the_cube * side_of_the_cube * side_of_the_cube;
13     cout << "V куба :" << v_of_the_cubs << endl;
14 }
```

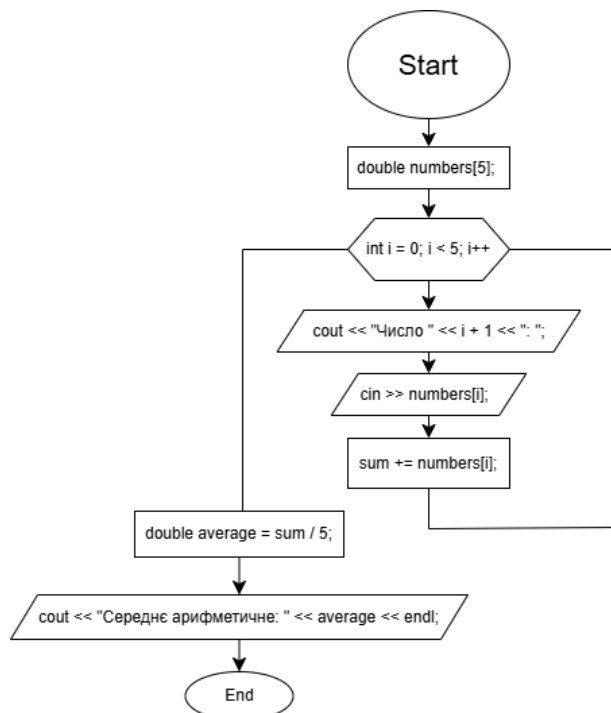


```
Введіть сторону куба: 6
V куба :216
```

## Завдання №4 VNS Practice Work - Task 4

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      double numbers[5];
7      double sum = 0.0;
8
9      cout << "Введіть 5 дробових чисел:" << endl;
10     for (int i = 0; i < 5; i++) {
11         cout << "Число " << i + 1 << ": ";
12         cin >> numbers[i];
13         sum += numbers[i];
14     }
15
16     double average = sum / 5;
17
18     cout << "Середнє арифметичне: " << average << endl;
19
20     return 0;
21 }
```

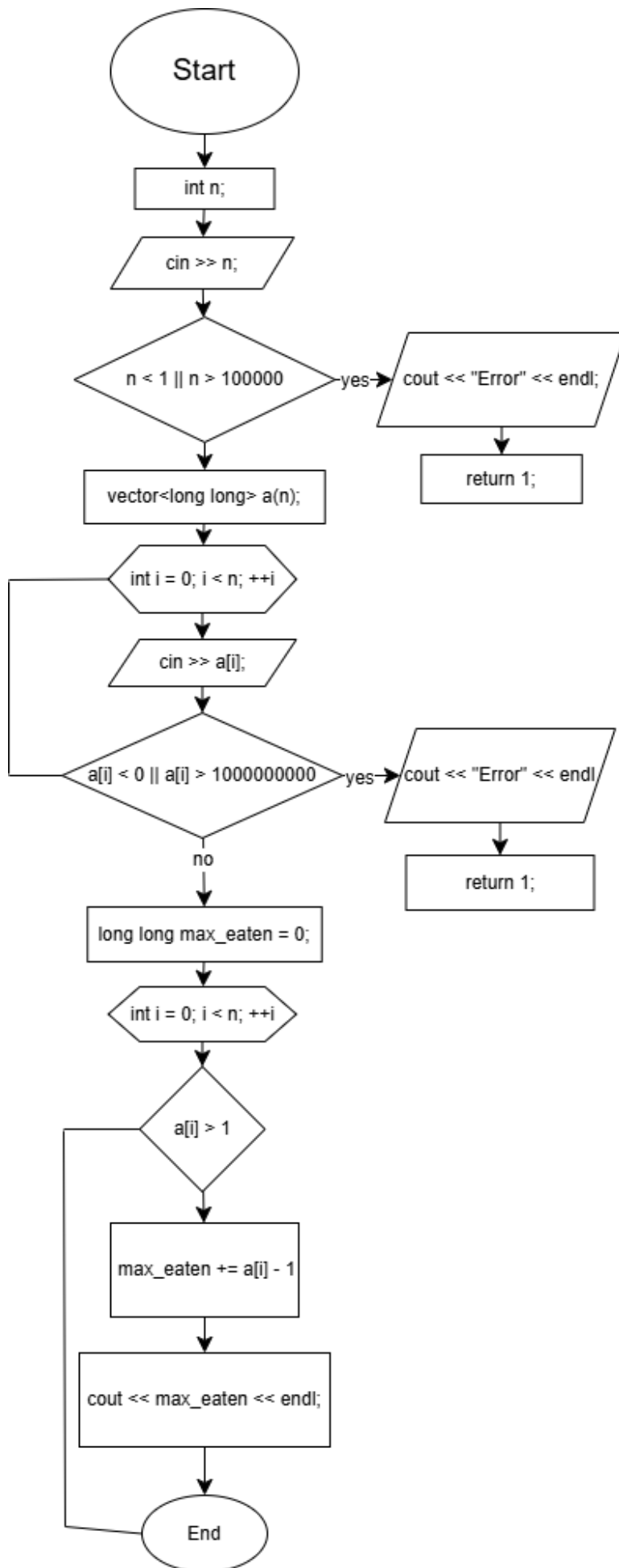
```
Введіть 5 дробових чисел:
Число 1: 4.5
Число 2: 3.2
Число 3: 1.2
Число 4: 4.7
Число 5: 8.8
Середнє арифметичне: 4.48
```



## Завдання №5 Algotester

```
1  #include <iostream>
2  #include <vector>
3  using namespace std;
4
5  int main() {
6      int n;
7      cin >> n;
8
9      if (n < 1 || n > 100000)
10     {
11         cout << "Error" << endl;
12         return 1;
13     }
14
15     vector<long long> a(n);
16     for (int i = 0; i < n; ++i)
17     {
18         cin >> a[i];
19
20         if (a[i] < 0 || a[i] > 1000000000)
21         {
22             cout << "Error" << endl;
23             return 1;
24         }
25     }
26
27     long long max_eaten = 0;
28     for (int i = 0; i < n; ++i) {
29         if (a[i] > 1) {
30             max_eaten += a[i] - 1;
31         }
32     }
33
34     cout << max_eaten << endl;
35     return 0;
36 }
37
```

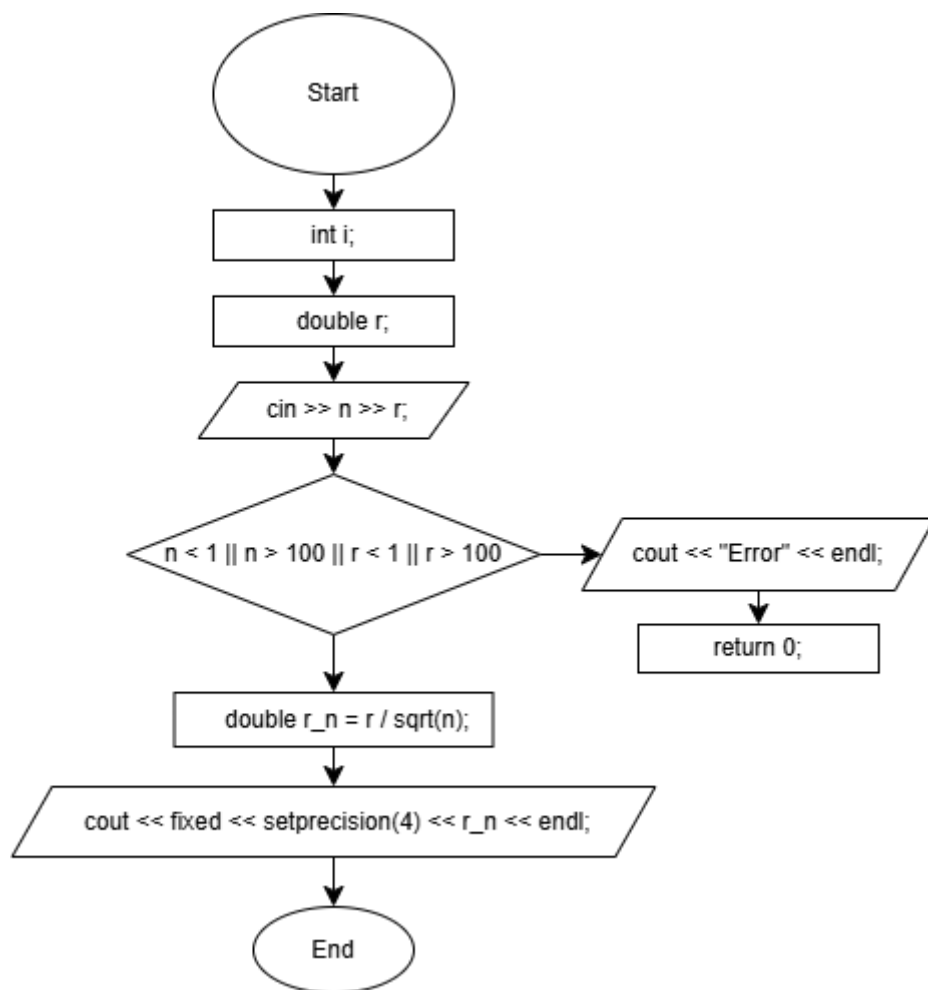
```
4
4 7 47 74
128
```



## Завдання №6 Algotester

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  int main()
7  {
8      int n;
9      cin >> n;
10
11     vector<pair<int, int>> offices(n);
12
13     for (int i = 0; i < n; i++) {
14         int length;
15         cin >> length;
16         offices[i] = {length, i + 1};
17     }
18
19     sort(offices.begin(), offices.end());
20
21     for (const auto& office : offices) {
22         cout << office.second << " ";
23     }
24     cout << endl;
25
26     return 0;
27 }
28
```

```
4 7
3.5000
```



## Завдання №7 Algotester

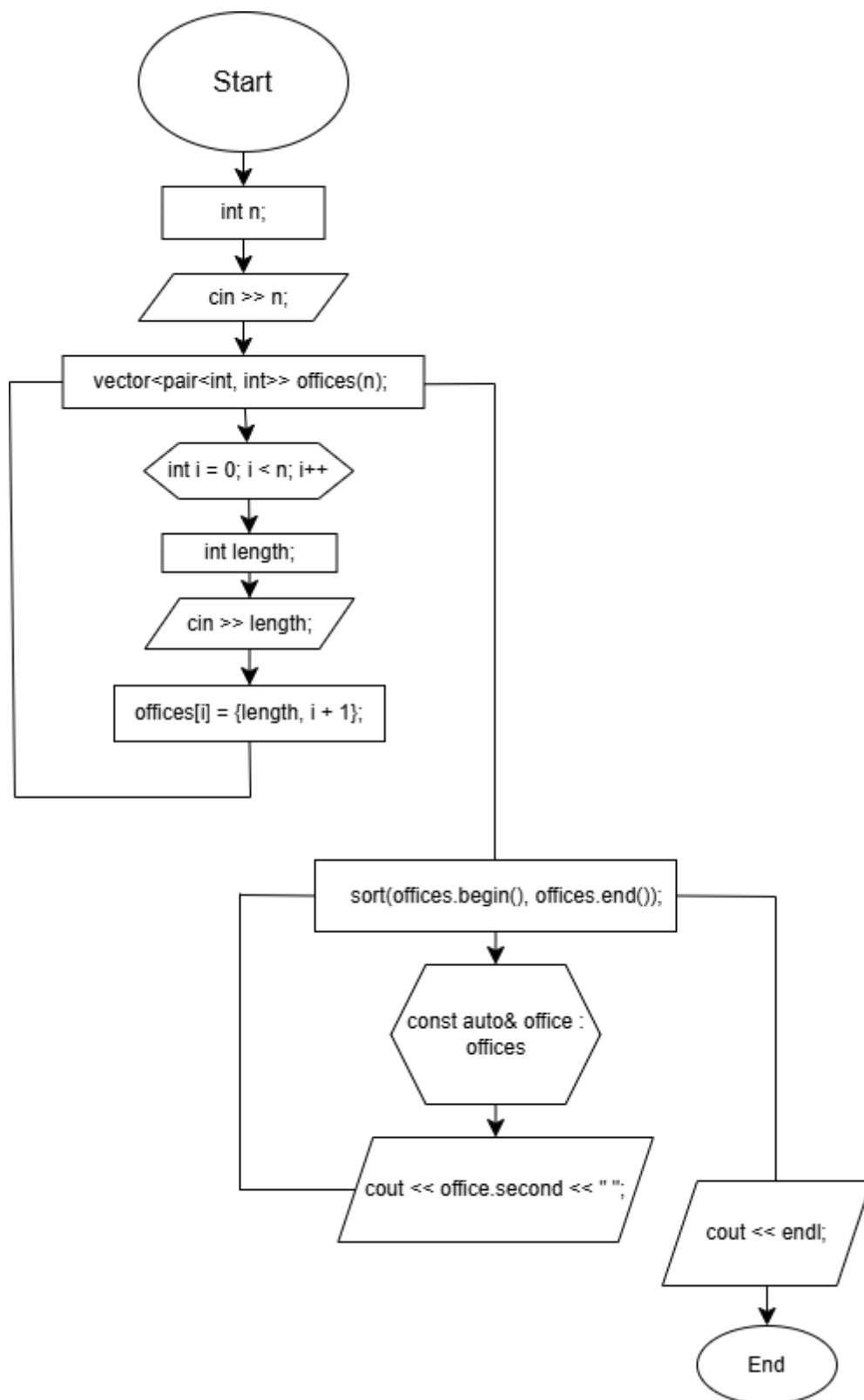
```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  int main()
7  {
8      int n;
9      cin >> n;
10
11     vector<pair<int, int>> offices(n);
12
13     for (int i = 0; i < n; i++) {
14         int length;
15         cin >> length;
16         offices[i] = {length, i + 1};
17     }
18
19     sort(offices.begin(), offices.end());
20
21     for (const auto& office : offices) {
22         cout << office.second << " ";
23     }
24     cout << endl;
25
26     return 0;
27 }
28
```

4

4 1 2 3

2 3 4 1

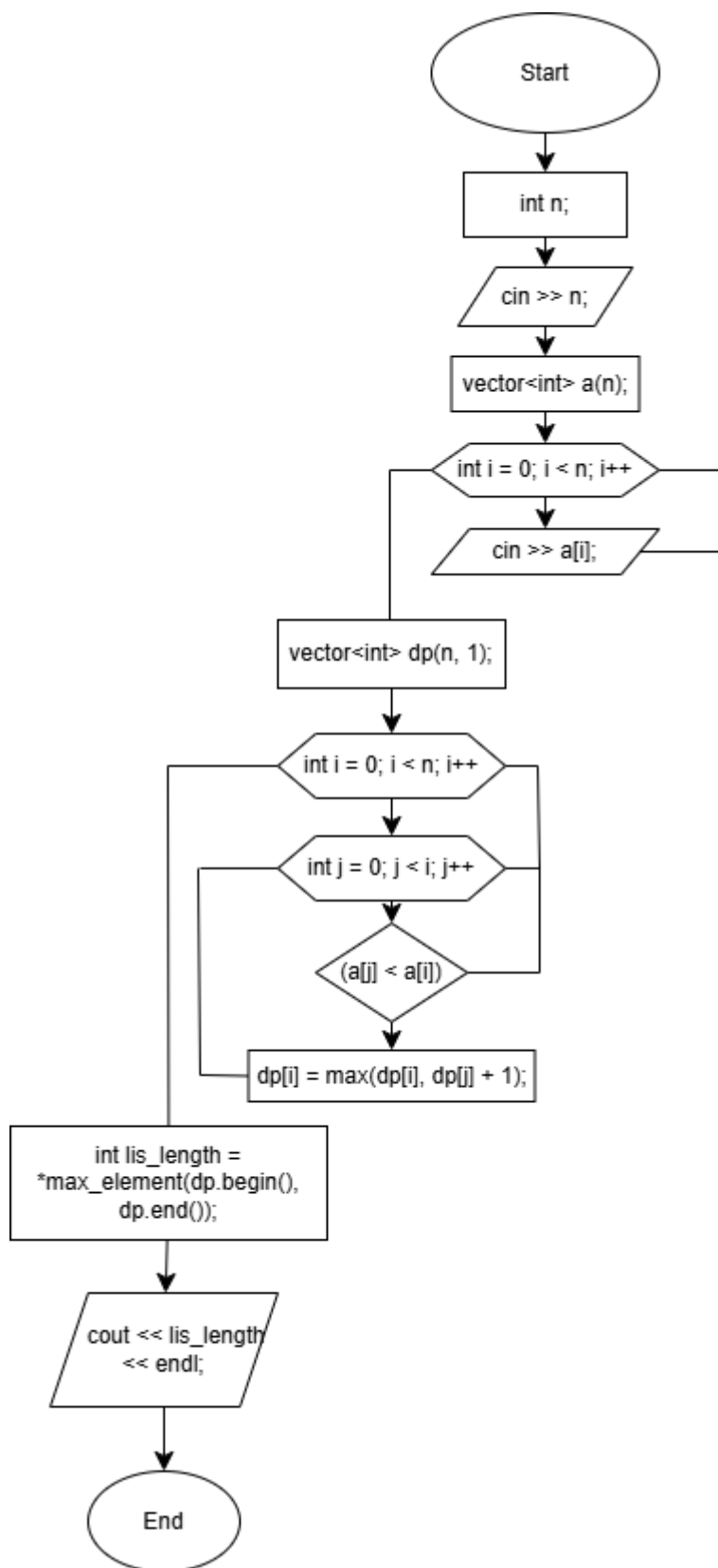




## Завдання №8 Algotester

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4  using namespace std;
5
6  int main()
7  {
8      int n;
9      cin >> n;
10
11     vector<int> a(n);
12     for (int i = 0; i < n; i++) {
13         cin >> a[i];
14     }
15
16     vector<int> dp(n, 1);
17
18     for (int i = 0; i < n; i++) {
19         for (int j = 0; j < i; j++) {
20             if (a[j] < a[i]) {
21                 dp[i] = max(dp[i], dp[j] + 1);
22             }
23         }
24     }
25
26     int lis_length = *max_element(dp.begin(), dp.end());
27
28     cout << lis_length << endl;
29
30     return 0;
31 }
32
33
```

```
5
4 7 2 10 3
3
```



***Висновок:*** зробивши цю розрахункову роботу (Сага1) , я закріпила всі свої знання ,які здобула за минулі лабораторні роботи.