

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:
Студент групи ШІ-12
Іванів Христина

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

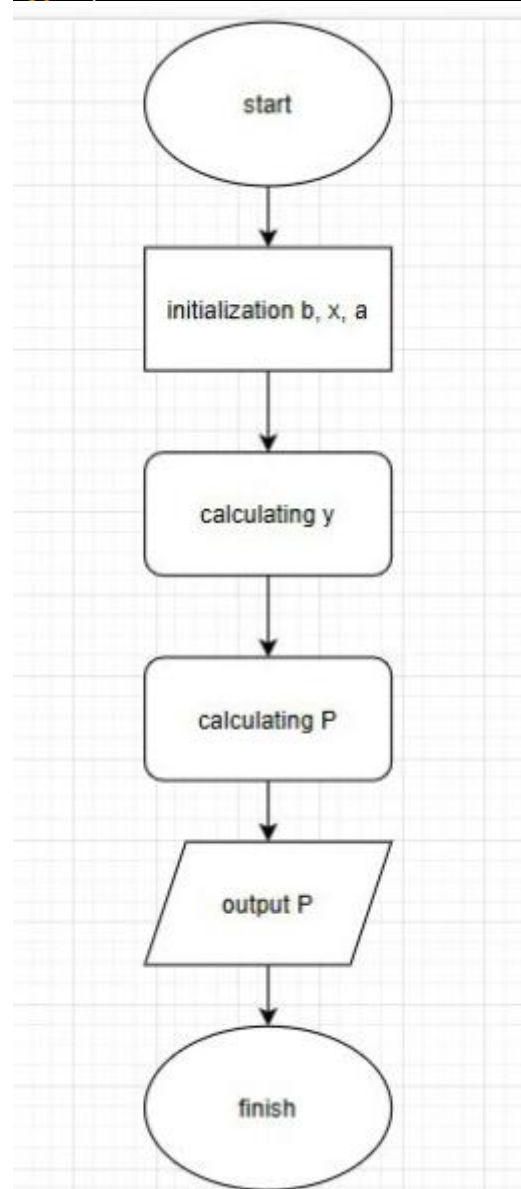
Завдання 1

Розробити лінійний алгоритм для розв'язання задачі:

Варіант 6. $P = \sqrt{x^2 + b} - b^2 - \sin^3(x + a)/xy$,

де $b = 15,5$; $x = -2,9$; $y = \cos^2 x^3 - \frac{x}{\sqrt{a^2 + b^2}}$; $a = 1,5$.

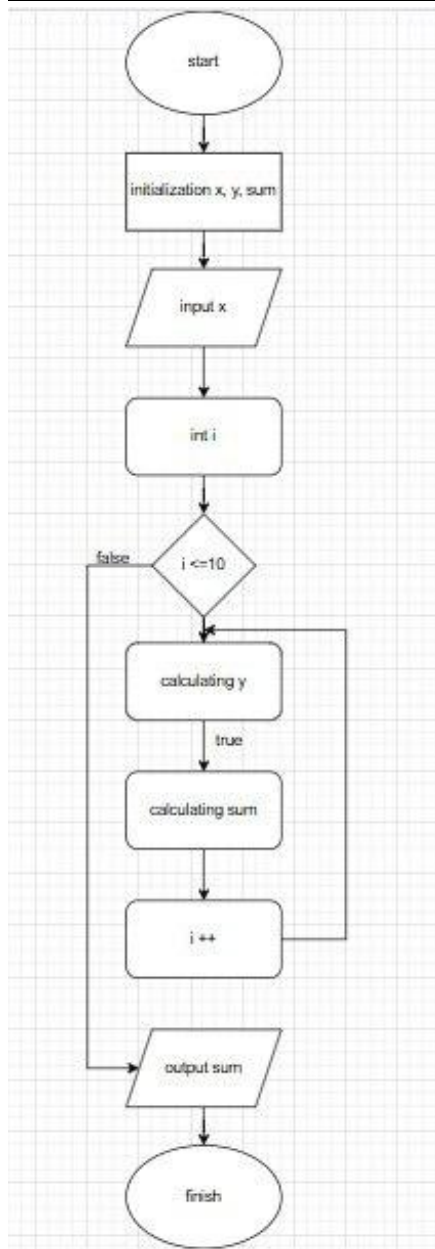
```
1  #include <math.h>
2  #include <iostream>
3  int main()
4  {
5
6      float b = 15.5;
7      float x = -2.9;
8      float a = 1.5;
9      double y = pow(cos(pow(x, 3)), 2) - x / (sqrt(pow(a, 2) + pow(b, 2)));
10     double P = sqrt(pow(x, 2) + b) - (pow(b, 2) * pow(sin(x + a), 3)) / (x * y);
11     std::cout << P;
12     return 0;
13 }
```



Завдання 2. Розробити алгоритм, що розгалужується для розв'язання задачі номер якої відповідає порядковому номеру студента в журналі викладача

Варіант 25. Обчислити функцію у вигляді формули $y = \sum_{i=1}^{10} \frac{x^i}{i}$ для кожного i.

```
1  #include <math.h>
2  #include <iostream>
3  int main()
4  {
5      double x, y;
6      double sum = 0;
7      std::cin >> x;
8      for (int i = 1; i <= 10; i++)
9      {
10         y = (pow(x, i)) / i;
11         sum += y;
12     }
13     std::cout << sum;
14     return 0;
15 }
```



Завдання 3. Написати програму згідно свого варіанту.

Варіант 20. Обчислює швидкість, з якою бігун пробіг дистанцію. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення швидкості бігу.

Введіть довжину дистанції (метрів) > **1000**

Ведіть час (хвилин.секунд) > **3.25**

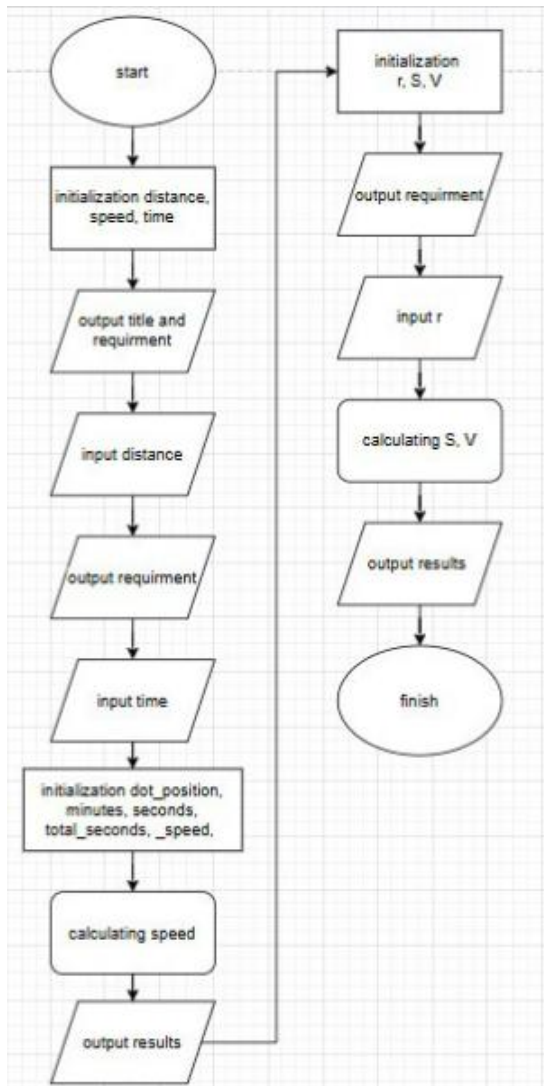
Дистанція: 1000 м

Часи 3 мін 25 сік = 205 сік

Швидкості бігу = 17.56 км/година

Обчислення об'єму $V = \frac{3}{4} \pi * r^3$ і площі поверхні кулі $S = 4 \pi * r^2$.

```
1  #include <iostream>
2  #include <string>
3  #include <iomanip>
4  #include <cmath>
5  using namespace std;
6
7  int main() {
8      double distance, speed;
9      string time;
10
11      cout << "Обчислення швидкості бігу." << endl;
12      cout << "Введіть довжину дистанції (метрів) = ";
13      cin >> distance;
14      cout << "Введіть час (хвилин.секунд) = ";
15      cin >> time;
16
17      size_t dot_position = time.find('.');
18
19      int minutes = stoi(time.substr(0, dot_position));
20      int seconds = stoi(time.substr(dot_position + 1));
21
22      int total_seconds = minutes * 60 + seconds;
23
24      double _speed = distance / total_seconds;
25
26      float speed = _speed*3.6;
27
28      cout << fixed << setprecision(2);
29      cout << "Дистанція: " << distance << " м" << endl;
30      cout << "Час " << minutes << " хв " << seconds << " сек = " << total_seconds << " сек" << endl;
31      cout << "Швидкість бігу = " << speed << " км/год" << endl;
32
33      double r, S, V;
34      cout << "Введіть радіус кулі: ";
35      cin >> r;
36
37      S = 4*M_PI*pow(r,2);
38      V = (3.0/4.0)*M_PI*pow(r,3);
39
40      cout << "S = " << S << endl;
41      cout << "V = " << V << endl;
42
43      return 0;
44  }
```



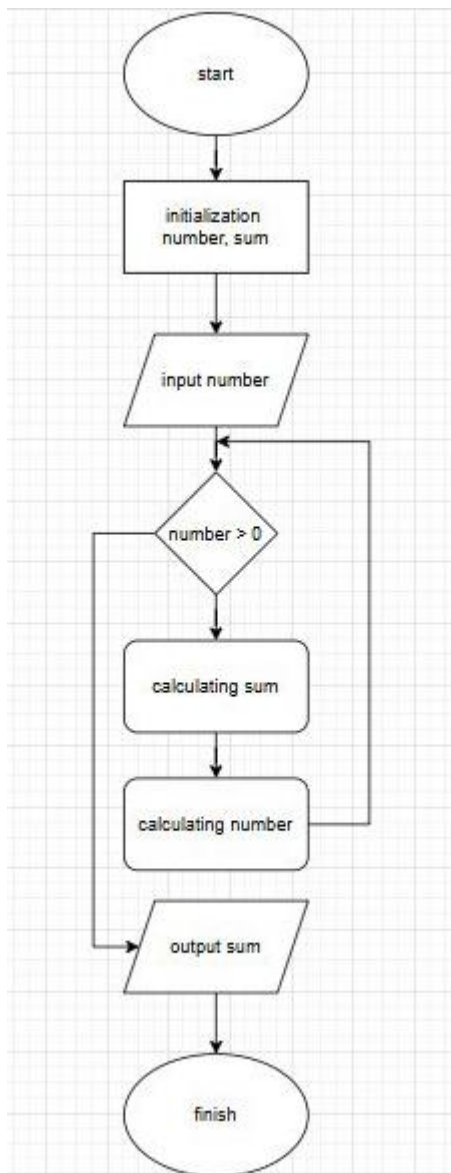
Завдання 4. Написати програму згідно свого варіанту.

Варіант 1. Знайти суму цифр введеного з клавіатури числа.

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int number, sum;
8      cin >> number;
9      while (number > 0)
10     {
11         sum += number % 10;
12         number /= 10;
13     }
14     cout << sum;
15     return 0;
16 }

```



Algotester task 1

Верховна Рада

Limits: 2 sec., 256 MiB

Вже не перший рік у нашому суспільстві гостро стоїть питання кількості народних депутатів у Верховній Раді. Утримувати дуже багато дармоїдів-депутатів народ не хоче, проте все має бути справедливо і кожна політична партія повинна отримати кількість місць у парламенті пропорційну до кількості голосів, що вона отримала на виборах. Більш формально, відношення кількості голосів до кількості місць у Верховній Раді має бути однаковим для усіх партій.

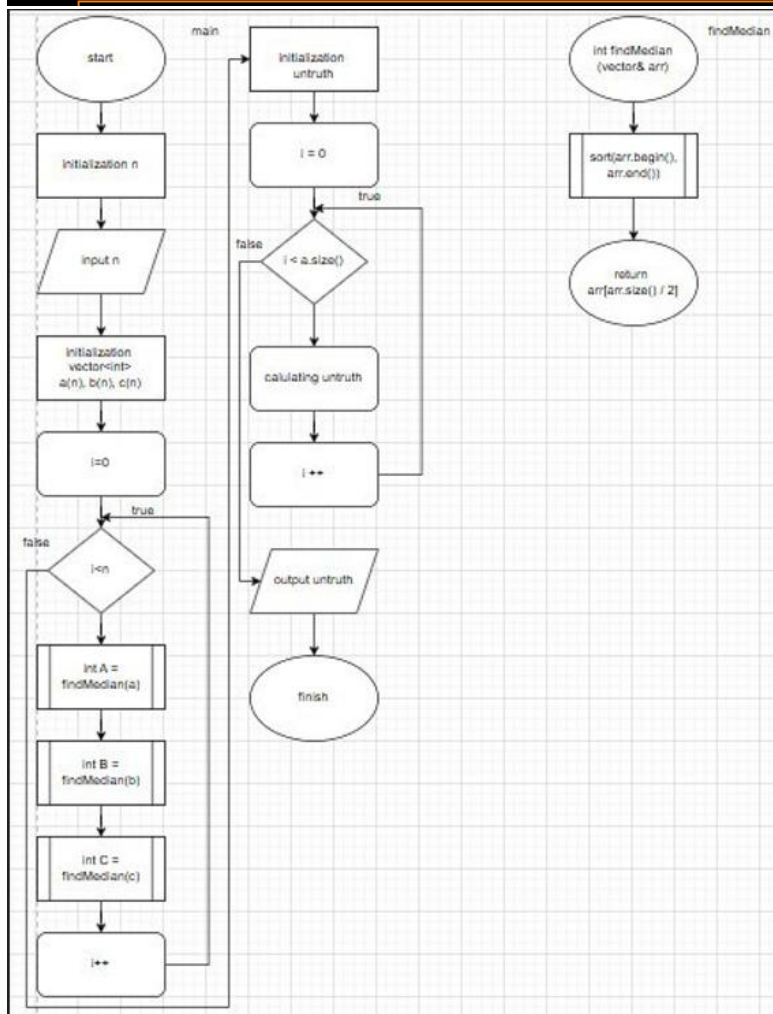
В останніх виборах до Верховної Ради взяли участь n партій. Вам відома кількість голосів, що отримала кожна з них. Допоможіть народові мінімізувати загальну кількість депутатських місць.

C++ 23	Accepted	0.003	1.199
--------	----------	-------	-------

```

1  #include <iostream>
2  #include <cmath>
3  #include<vector>
4  #include<algorithm>
5  using namespace std;
6
7  int findMedian(vector<int>& arr)
8  {
9      sort(arr.begin(), arr.end());
10     return arr[arr.size() / 2];
11 }
12
13 int main(){
14     int n;
15     cin >> n;
16
17     vector<int> a(n), b(n), c(n);
18
19     for (int i = 0; i < n; i++){
20         cin >> a[i] >> b[i] >> c[i];
21     }
22     int A = findMedian(a);
23     int B = findMedian(b);
24     int C = findMedian(c);
25
26     int untruth = 0;
27     for (unsigned int i = 0; i < a.size(); i++)
28     {
29         untruth += abs(A - a[i]) + abs(B - b[i]) + abs(C - c[i]);
30     }
31
32     cout << untruth;
33     return 0;
34 }

```



Algotester task 2

Борщ, картопля і салат

Limits: 2 sec., 256 MiB

Сьогодні в пластунів свято! Привезли вдвічі більше продуктів на обід. Зеник разом із Марічкою зголосилися допомагати на кухні. Всього є три різні страви: борщ, картопля і салат.

У Пласті в Зеника та Марічки є n друзів. i -й друг хоче з'їсти a_i грамів борщу, b_i грамів картоплі та c_i грамів салату. Якщо друг Зеника та Марічки хоче з'їсти x грамів певної страви, а йому в тарілку поклали y грамів, то ступінь недовіри до Зеника з Марічкою в цього друга зростає на $|x - y|$ одиниць. Зенику з Марічкою наказали накладати всім однакові порції певної страви. Тобто кожному другу потрібно накласти A грамів борщу, B грамів картоплі та C грамів салату.

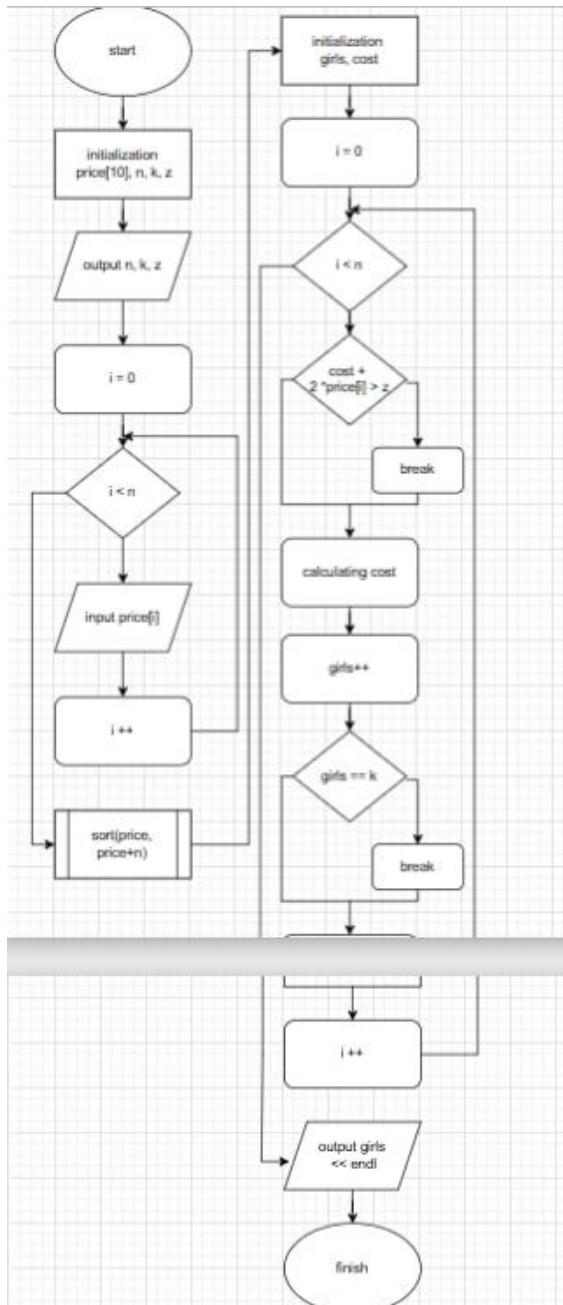
Допоможіть Зенику та Марічці обрати такі A , B та C , щоб сумарний ступінь недовіри всіх друзів був якомога меншим.

C++ 23	Accepted	0.108	2.289
--------	----------	-------	-------

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int gcd(int a, int b)
8  {
9      while (b != 0)
10     {
11         int temp = b;
12         b = a % b;
13         a = temp;
14     }
15     return a;
16 }
17
18
19 int main()
20 {
21     int n;
22     cin >> n;
23     vector<int> a(n);
24
25     for (int i = 0; i < n; i++)
26     {
27         cin >> a[i];
28     }
29
30     int result = a[0];
31
32     for (int i = 1; i < n; i++)
33     {
34         result = gcd(result, a[i]);
35     }
36
37     long long amount = 0;
38     for (int i = 0; i < n; i++)
39     {
40         if(result != 0)
41             amount += a[i] / result;
42     }
43     cout << amount;
44
45     return 0;
46 }
47
```



```
1  #include <iostream>
2  #include <algorithm>
3
4  using namespace std;
5
6  int main() {
7      int price[10], n, k, z;
8
9      cin >> n >> k >> z;
10
11     for (int i = 0; i < n; i++) {
12         cin >> price[i];
13     }
14
15     sort(price, price + n);
16     int girls = 0;
17     int cost = 0;
18
19     for (int i = 0; i < n; i++) {
20         if (cost + 2 * price[i] > z) {
21             break;
22         }
23         cost += 2 * price[i];
24         girls++;
25         if (girls == k) {
26             break;
27         }
28     }
29
30     cout << girls << endl;
31
32     return 0;
33 }
```



Algotester task 4

Дуби та берези

Limits: 2 sec., 256 MiB

Усі ви знаєте, що студентам часто доручають надзвичайно важливу роботу, яку просто нікому виконати. Оте приберіть, оце зберіть, туди сходіть, звідти прийдіть...

От і цього разу студентів факультету прикладної математики попросили порахувати кількість дерев у парку, що біля гуртожитку.

Студенти, котрі шойно відсвяткували День програміста, з радістю погодились виконати це цікаве завдання.

Кожен порахував кількість дерев кожного виду у парку.

Біда в тому, що молодь діяла не надто узгоджено, тому різні студенти могли врахувати в свою статистику ті самі дерева. Звісно ж, дехто вважав зайвим рахувати всі дерева, і тому врахував лише деякі. Але ми впевнені, що усі студенти безпомилково визначають вид дерева та завжди пам'ятають, котрі дерева вони вже врахували.

Відомо, що завдання виконували n студентів, і кожен із них зібрав дані про m видів дерев.

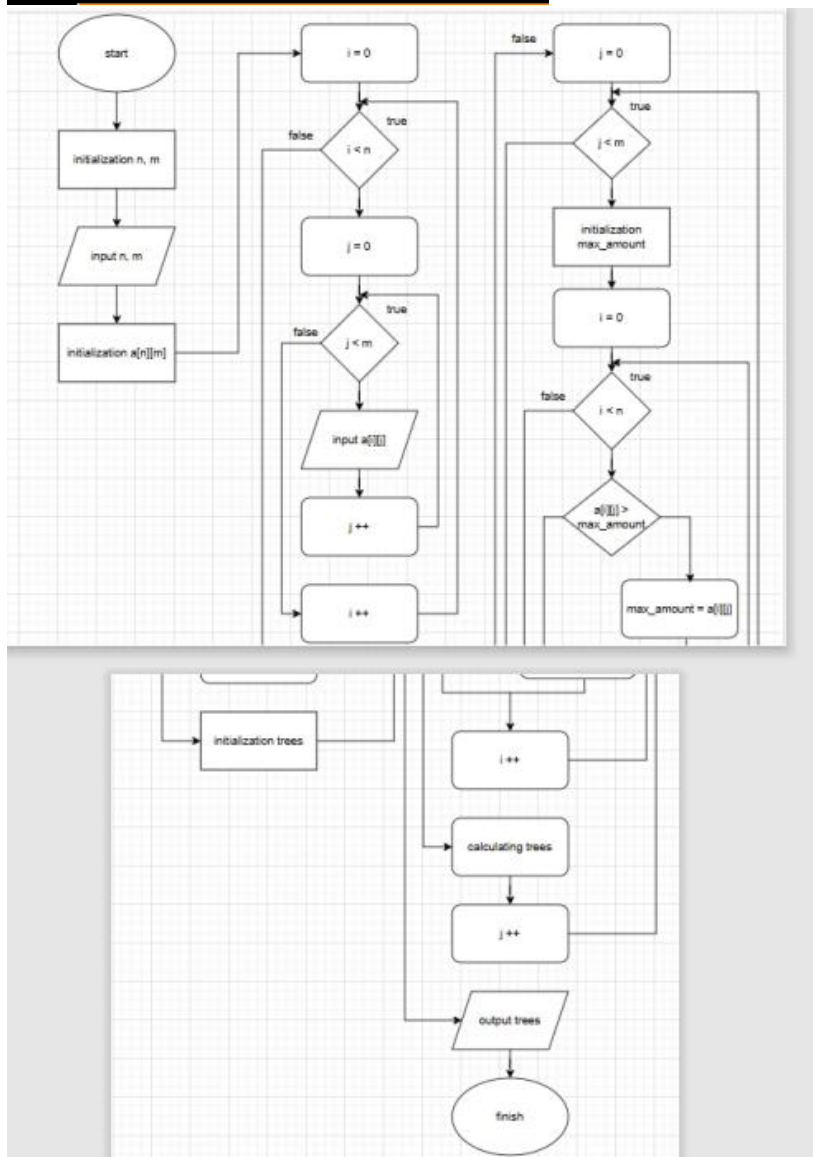
Вам потрібно допомогти адміністрації гуртожитку — напишіть програму, котра визначить мінімальну можливу кількість дерев в парку.

C++ 23	Accepted	0.003	1.168
--------	----------	-------	-------

```

1  #include <iostream>
2  using namespace std;
3  int main()
4  {
5      int n, m;
6      cin >> n >> m;
7      int a[n][m];
8      for (int i = 0; i < n; i++)
9      {
10         for (int j = 0; j < m; j++)
11         {
12             cin >> a[i][j];
13         }
14     }
15
16     int trees = 0;
17     for (int j = 0; j < m; j++)
18     {
19         int max_amount = -1;
20         for (int i = 0; i < n; i++)
21         {
22             if (a[i][j] > max_amount)
23                 max_amount = a[i][j];
24         }
25         trees += max_amount;
26     }
27
28     cout << trees;
29     return 0;
30 }

```



Висновок:

Під час виконання саги я закріпила навички використання циклів, умовних операторів, функцій вводу та виводу, масивів, змінних та констант