

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**

Виконав:

Студент групи ІІІ-12
Бобровицький Олександр Сергійович

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

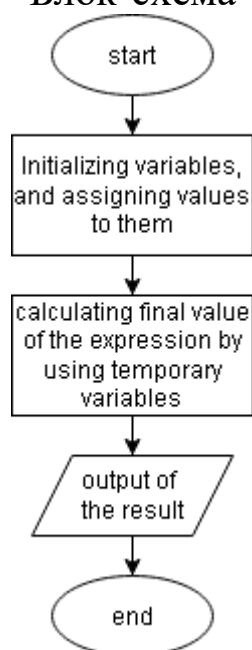
Завдання 1

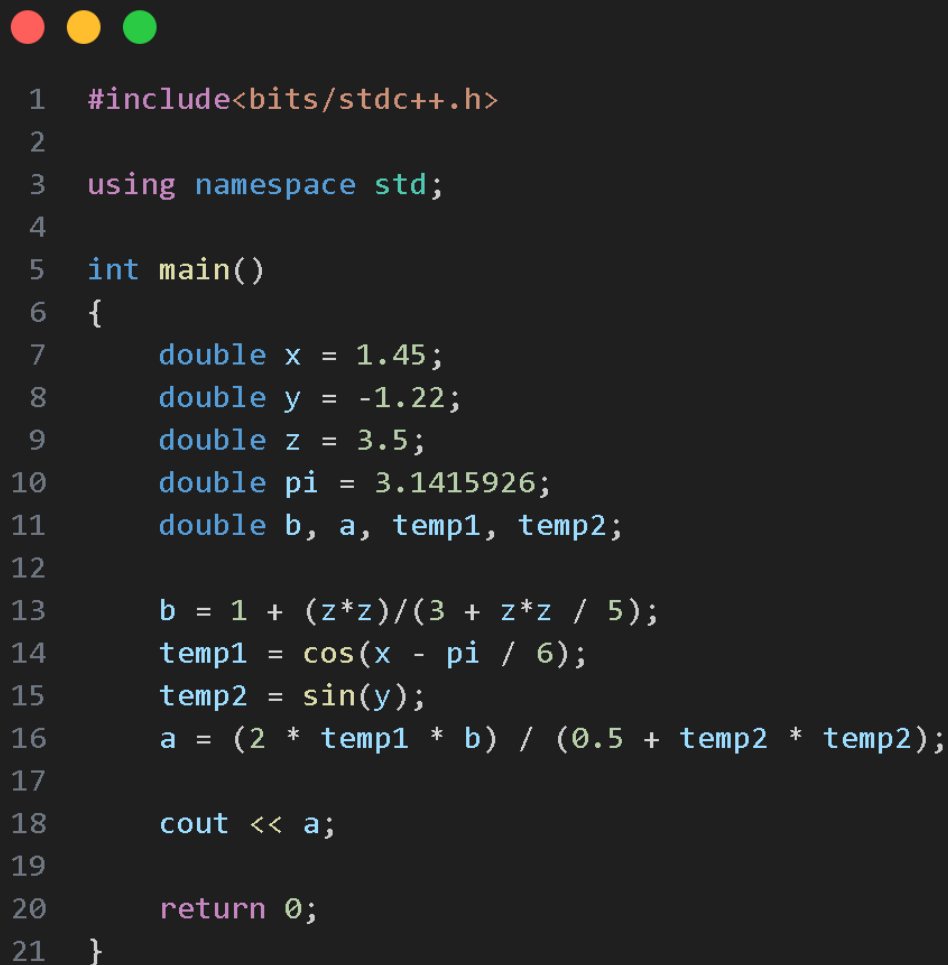
Розробити лінійний алгоритм для розв'язання задачі.

Варіант 2. $a = \frac{2 \cos(x - \frac{\pi}{6})b}{\frac{1}{2} + \sin^2 y}$; $b = 1 + \frac{z^2}{3 + z^2 / 5}$, де $x=1,45$;

$y=-1,22$; $z=3,5$.

Блок-схема





```

1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      double x = 1.45;
8      double y = -1.22;
9      double z = 3.5;
10     double pi = 3.1415926;
11     double b, a, temp1, temp2;
12
13     b = 1 + (z*z)/(3 + z*z / 5);
14     temp1 = cos(x - pi / 6);
15     temp2 = sin(y);
16     a = (2 * temp1 * b) / (0.5 + temp2 * temp2);
17
18     cout << a;
19
20     return 0;
21 }

```

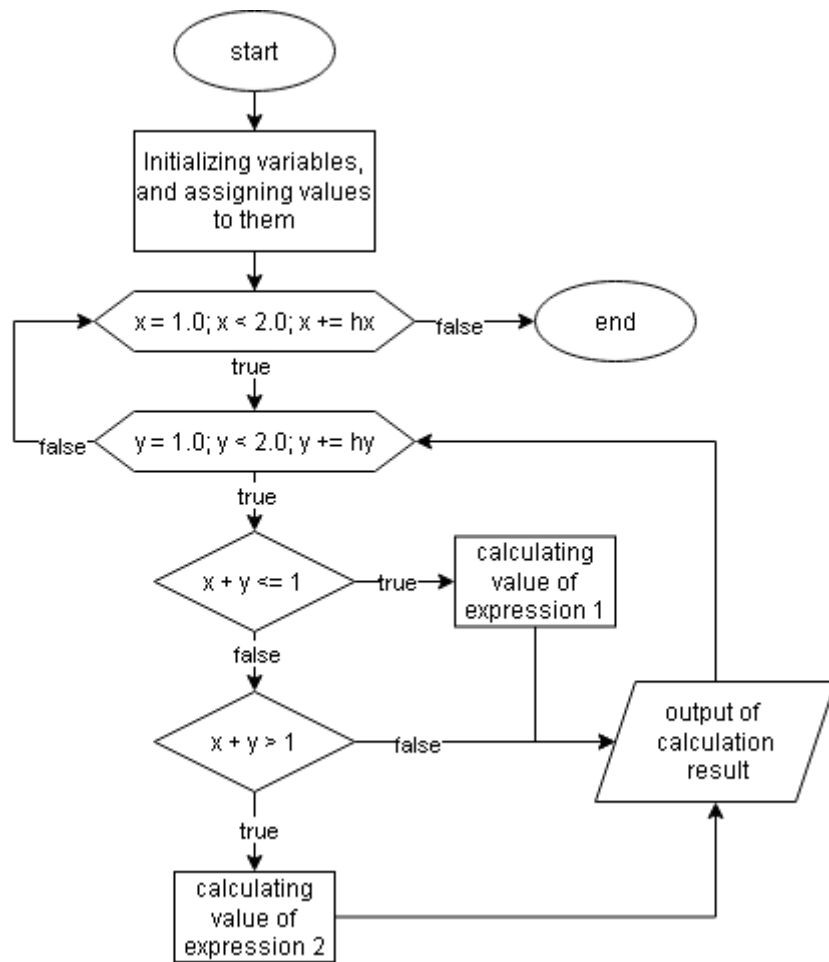
Завдання 2


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

$$\text{Варіант 2. } z = \begin{cases} a^x + b^y, & x + y \leq 1, \\ ax^2 + \ln(bxy), & x + y > 1; \end{cases}$$

де $x \in [1, 2]$; $h_x = 0,1$; $a = 1$; $b = 2$; $y \in [1, 2]$; $h_y = 0,2$.

Блок-схема





```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      double hx = 0.1;
8      double hy = 0.2;
9      int a = 1;
10     int b = 2;
11     double z;
12
13     for(double x = 1.0; x < 2.0; x += hx)
14     {
15         for(double y = 1.0; y < 2.0; y += hy)
16         {
17             if (x + y <= 1)
18             {
19                 z = pow(a, x) + pow(b, y);
20             }
21             else if (x + y > 1)
22             {
23                 z = a*x*x + log(b * x * y);
24             }
25
26             cout << z << endl;
27         }
28     }
29
30     return 0;
31 }
```

Завдання 3

Варіант 2. Обчислення площі поверхні паралелепіпеда. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення площі поверхні паралелепіпеда.

Введіть початкові дані:

Довжина (см) > **9**

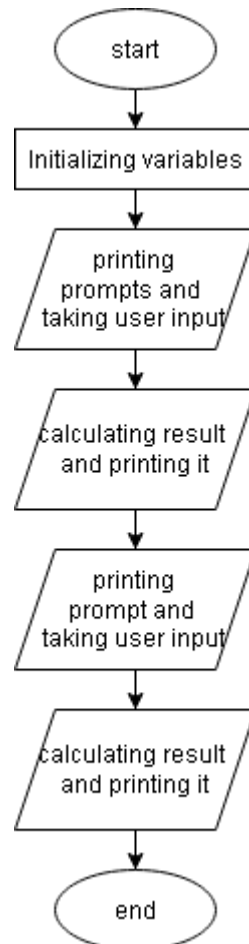
Ширина(см)> **7.5**

Висота(см)> **5**

Площа поверхні: 90.00 кв.см.

Реалізувати перерахунок відстані з кілометрів у версти (одна верста – це 1066,8 м).

Блок-схема



```

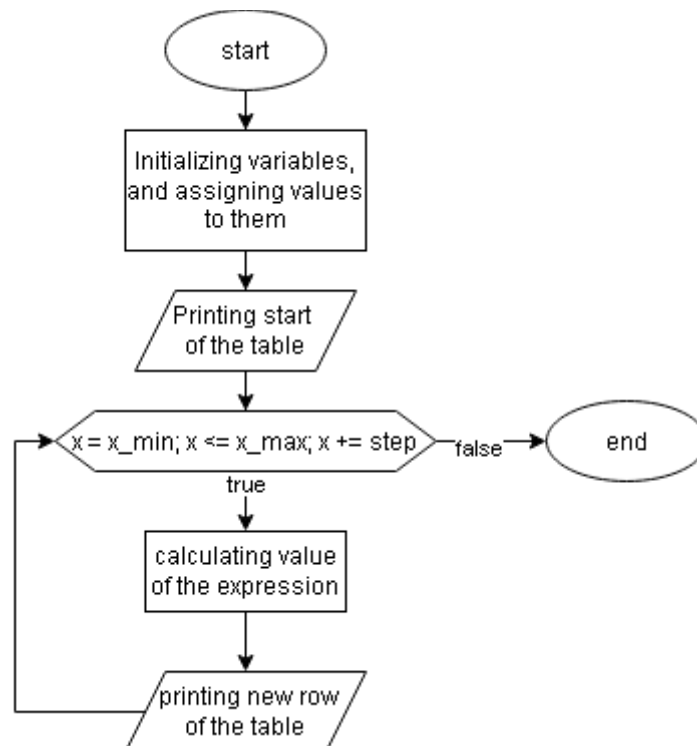
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      double length, width, height, km;
8      cout << "Обчислення площі поверхні паралелепіпеда\nВведіть початкові дані: \n";
9      cout << "Довжина (см) - ";
10     cin >> length;
11     cout << "Ширина(см) - ";
12     cin >> width;
13     cout << "Висота(см) - ";
14     cin >> height;
15
16     cout << "Площа поверхні: " << 2*(length + width) * height + 2*(length * width)<< " кв.см.\n";
17
18     cout << "Введіть відстань в кілометрах - ";
19     cin >> km;
20
21     cout << "Ця відстань в верстах - " << km / 1.0668;
22
23     return 0;
24 }

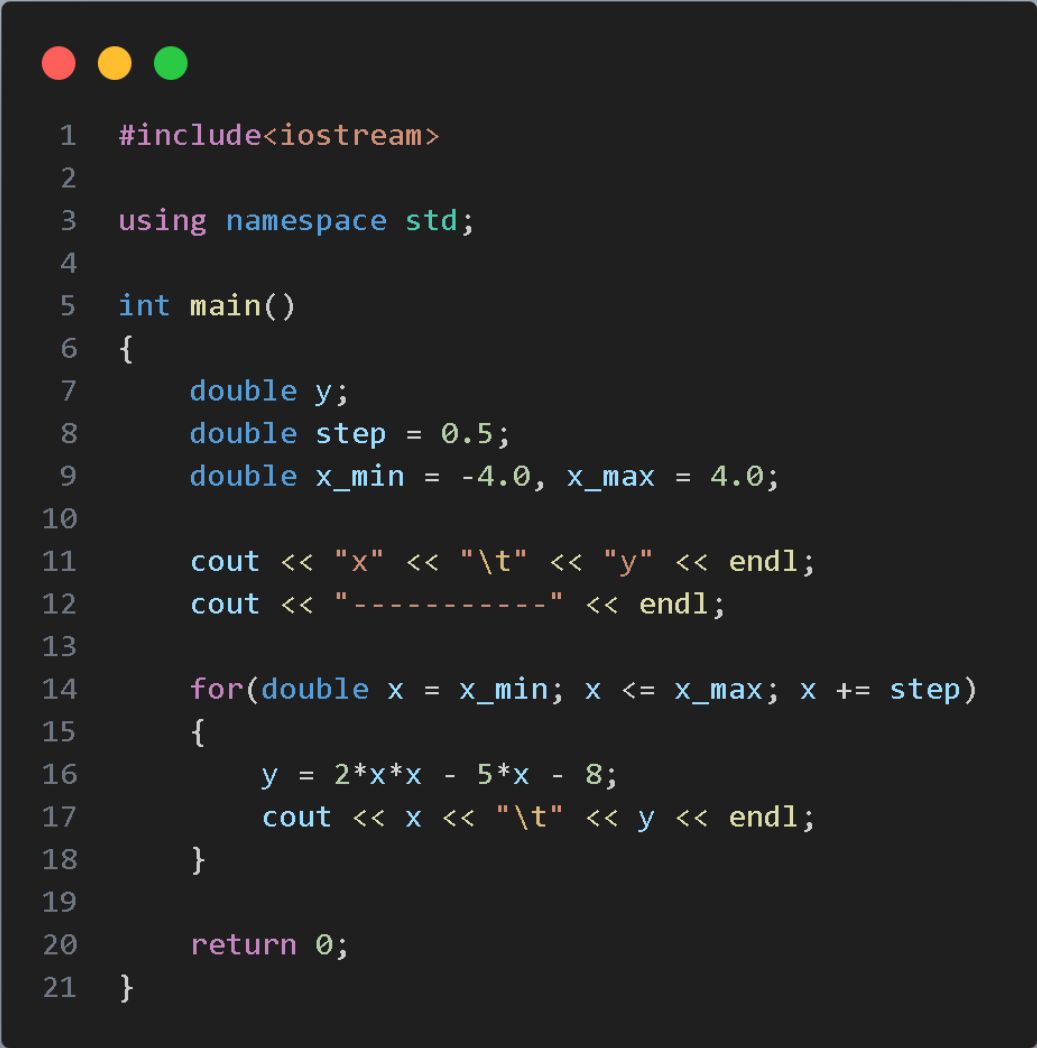
```

Завдання 4

Варіант 2. Напишіть програму, яка виводить на екран таблицю значень функції $y = 2x^2 - 5x - 8$ в діапазоні від -4 до 4. Крок зміни аргументу 0,5.

Блок-схема





```

1  #include<iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      double y;
8      double step = 0.5;
9      double x_min = -4.0, x_max = 4.0;
10
11     cout << "x" << "\t" << "y" << endl;
12     cout << "-----" << endl;
13
14     for(double x = x_min; x <= x_max; x += step)
15     {
16         y = 2*x*x - 5*x - 8;
17         cout << x << "\t" << y << endl;
18     }
19
20     return 0;
21 }

```

Завдання 5(0181 algotester)

Little Imp and Dragon like to spend their spare time together. Today they are playing an interesting game.

The game is played on N by M board. Initially all board cells are white. The players alternate turns and Little Imp starts the game. While making a turn a player selects a white board cell and paints it black. A player who can not make a valid move (because all the board is black) loses the game and his opponent is considered to be a winner.

Your task is to determine the game winner.

Input

Two integers N and M separated by a space.

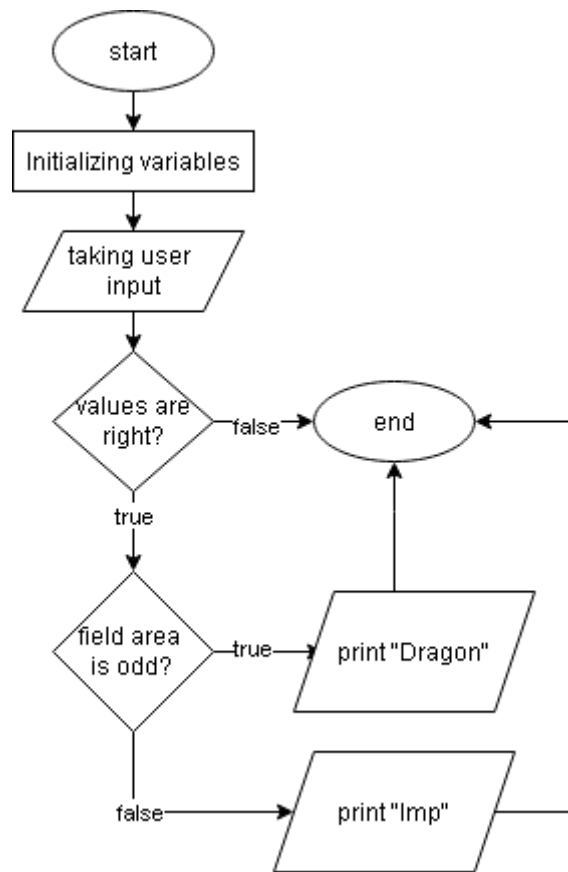
Output

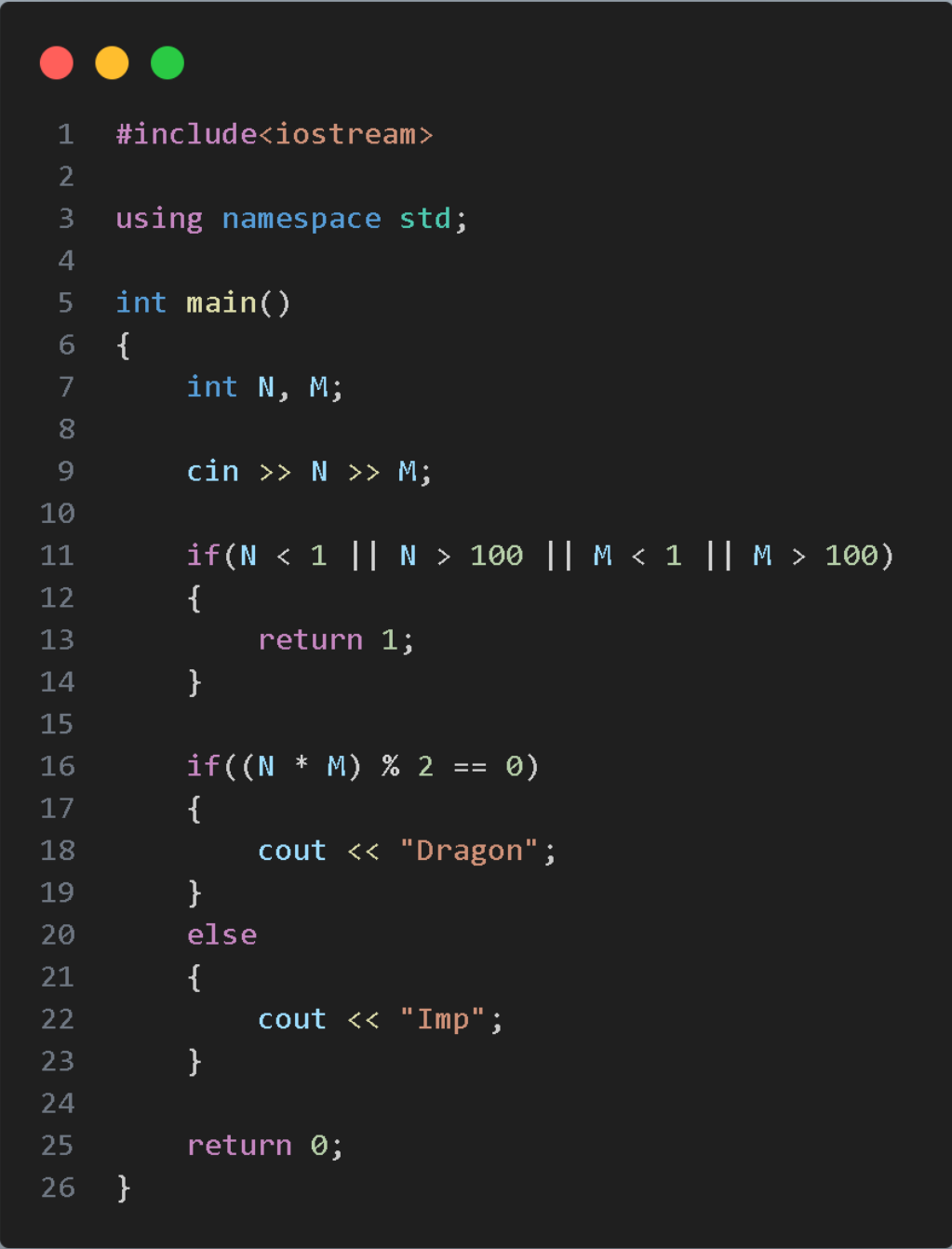
Print "Imp" (quotes for clarity) if Little Imp wins the game otherwise print "Dragon".

Constraints

$1 \leq N, M \leq 100$.

Блок-схема





```
1  #include<iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int N, M;
8
9      cin >> N >> M;
10
11     if(N < 1 || N > 100 || M < 1 || M > 100)
12     {
13         return 1;
14     }
15
16     if((N * M) % 2 == 0)
17     {
18         cout << "Dragon";
19     }
20     else
21     {
22         cout << "Imp";
23     }
24
25     return 0;
26 }
```

Завдання 6(0211 algotester)

Нарешті ми його дочекалися, 256-го дня в році (дня програміста)... Кожен зі студентів факультету прикладної математики та інформатики святкували його по-різному: дехто вдома, дехто в «Picasso», дехто в гуртожитку... Не дивно, що Зеник із Марічкою залишилися святкувати його в гуртожитку. Вони організували «mega party» :-). Свято вдалося...

Наступного дня, гуляючи з Марічкою, Зенику стало цікаво, скільки ж було випито різної випивки?!

Марічка змогла згадати n назв випивок, а Зеник — m .

Вам потрібно написати програму, яка порахує скільки ж було різної випивки на святі, яку змогли згадати Зеник з Марічкою.

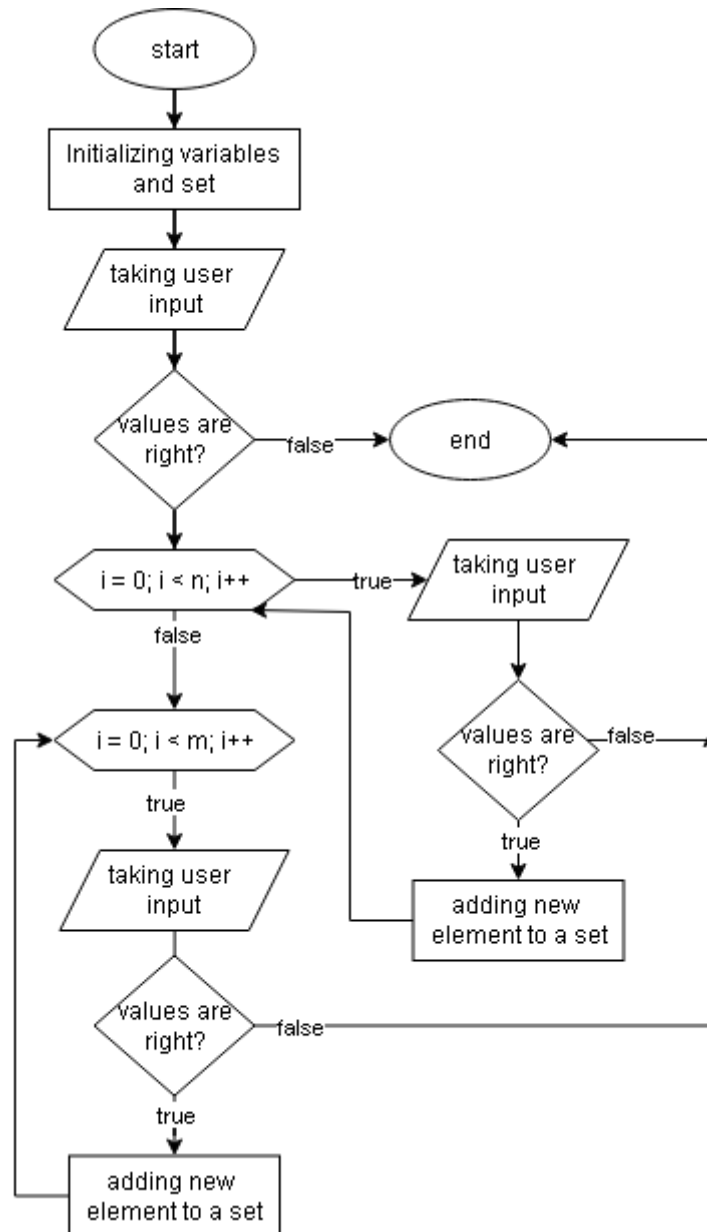
Input

У першому рядку дано цілі числа n і m .

У наступних n рядках дано назви випивок, які змогла згадати Марічка.

В наступних m рядках дано назви випивок, які зміг згадати Зеник.

Блок-схема





```
1  #include <bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int n, m;
8      string name;
9      unordered_set<string> drinks;
10
11     cin >> n >> m;
12
13     if(n < 0 || n > 777 || m < 0 || m > 777)
14     {
15         return 1;
16     }
17
18     for(int i = 0; i < n; i++)
19     {
20         cin >> name;
21         if(name.size() > 7 || name.size() < 1)
22         {
23             return 1;
24         }
25         drinks.insert(name);
26     }
27
28     for(int i = 0; i < m; i++)
29     {
30         cin >> name;
31         if(name.size() > 7 || name.size() < 1)
32         {
33             return 1;
34         }
35
36         drinks.insert(name);
37     }
38
39     cout << drinks.size() << endl;
40
41     return 0;
42 }
```

Завдання 7(0691 algotester)

Нещодавно ми мали змогу спостерігати за рідкісним явищем. Блакитний кривавий супермісяць — однозначно незабутнє видовище. Звісно ж, наш давній знайомий романтик Зеник не міг пропустити таку нагоду вразити Марічку. Тож він запросив її додому й розповідав про дивовижний сюрприз, який чекає на неї вночі.

Проте Марічка вперто не хотіла залишатися на ніч. І Зеник придумав, як її зацікавити. Він знайшов колоду з n карт. На кожній карті була записана мала літера латинського алфавіту. Завдання Марічки — скласти з наявних карт найдовший можливий паліндром.

Паліндром — це рядок, який однаково читається зліва направо та справа наліво. Наприклад, рядки `a`, `aba`, `рwкp` є паліндромами, водночас стрічки `ab`, `abc`, `aabb` — ні.

Input

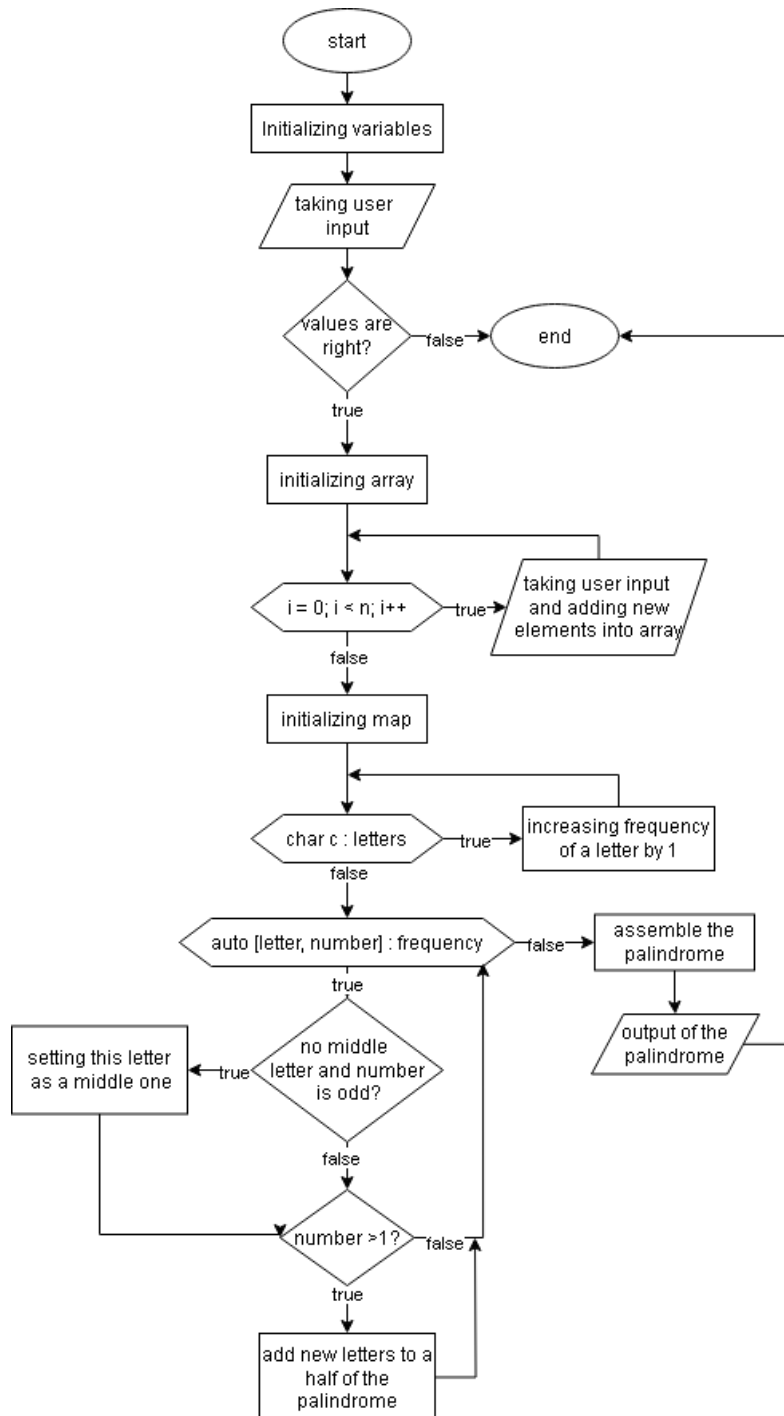
У першому рядку задано ціле число n — кількість карт.

У другому рядку задано n малих літер латинського алфавіту c_i — літера, написана на i -й карті.

Output

Виведіть один рядок — найдовший паліндром, який можна скласти з літер, написаних на картах. Якщо таких рядків декілька, виведіть будь-який.

Блок-схема





```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int n;
8      string palindrome, half, middle;
9
10     cin >> n;
11     if(n > 1000 || n < 1)
12     {
13         return 1;
14     }
15
16     vector<char> letters(n);
17     for(int i = 0; i < n; i++)
18     {
19         cin >> letters[i];
20     }
21
22     unordered_map<char, int> frequency;
23     for(char c : letters)
24     {
25         frequency[c]++;
26     }
27
28     for(auto [letter, number] : frequency)
29     {
30         if(middle.empty() && number % 2 != 0)
31         {
32             middle = letter;
33         }
34
35         if(number > 1)
36         {
37             half += string(number / 2, letter);
38         }
39     }
40
41     palindrome = half + middle;
42     reverse(half.begin(), half.end());
43     palindrome += half;
44
45     cout << palindrome;
46
47     return 0;
48 }
```

Завдання 8(0877 algotester)

Одного разу Дмитрик знайшов пристрій, який при натисканні кнопки «Показати» показує на екрані якесь ціле додатне число. Кожного разу, коли він бачить на екрані число менше або рівне за усі попередні числа (або перше число) він вигукує «Bay!».

Відомо, що Дмитрик натискав кнопку «Показати» n раз. Скільки раз він вигукнув «Bay!»?

Input

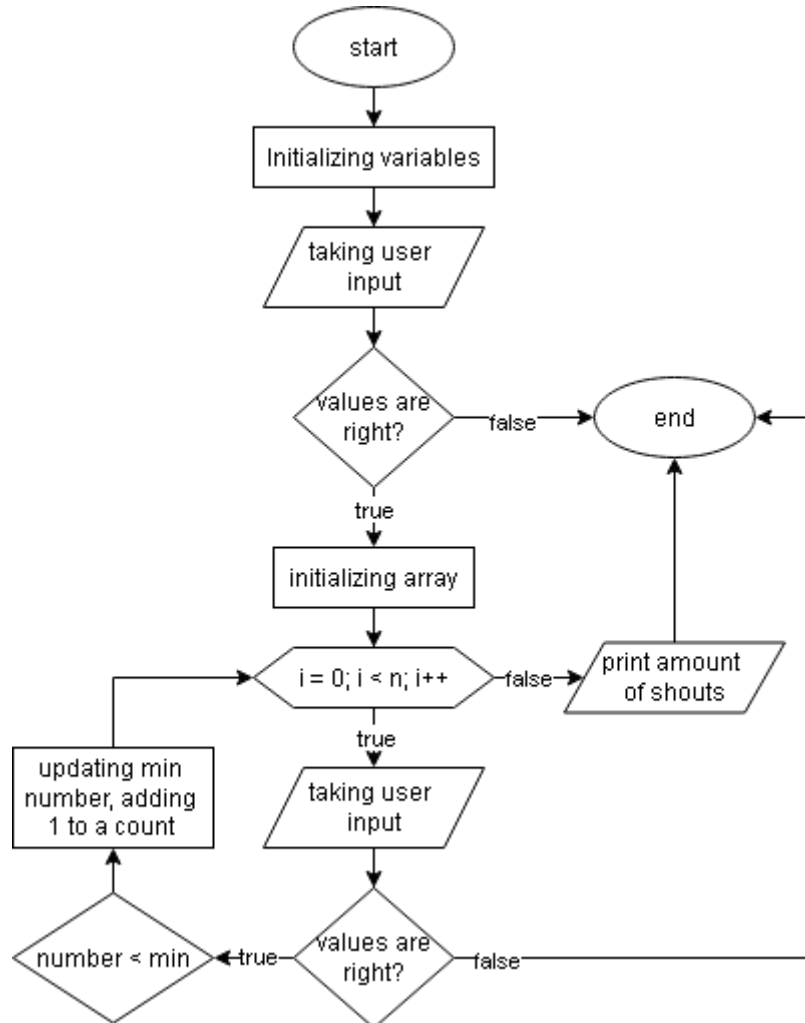
У першому рядку задано одне ціле число n — кількість натискань на кнопку.

У другому рядку задано n цілих чисел у тій послідовності, в якій вони появлялися на екрані в Дмитрика.

Output

У єдиному рядку виведіть одне ціле число — кількість разів коли Дмитрик вигукнув "Bay!"

Блок-схема





```
1  #include<bits/stdc++.h>
2
3  using namespace std;
4
5  int main()
6  {
7      int n,count = 0;
8      int min = INT_MAX;
9
10     cin >> n;
11     if(n > 1e6 || n < 1)
12     {
13         return 1;
14     }
15
16     vector<int> numbers(n);
17
18     for(int i = 0; i < n; i++)
19     {
20         cin >> numbers[i];
21         if(numbers[i] > 1e9 || numbers[i] < 1)
22         {
23             return 1;
24         }
25
26         if(numbers[i] <= min)
27         {
28             min = numbers[i];
29             count++;
30         }
31     }
32
33     cout << count;
34
35     return 0;
36 }
```


0877 - Мінімальні виходи	C++ 23	Accepted	
0877 - Мінімальні виходи	C++ 23	Wrong Answer 2	
0691 - Гра в карти	C++ 23	Accepted	
0211 - День програміста	C++ 23	Accepted	
0211 - День програміста	C++ 23	Wrong Answer 5	
0181 - Interesting Game	C++ 23	Accepted	

Висновок: я одержав практичні навички в розробці і дослідженні алгоритмів розв'язання задач.