

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(ла):**

Студент групи ШІ-12

Соснило Богдан Тарасович

Львів 2024

**Тема:** перевірка знань здобутих за цей семестр, а саме: функцій вводу та виводу, циклів (з умовою, з передумовою та з після умовою), умовних операторів, одновимірних та двовимірних масивів, створення та ініціалізація змінних та констант, створення та виклик функції, перенавантаження функції, рекурсивна функція, використання структур, виконання унарних та бінарних операцій.

**Мета:** перевірити себе на знання з вище наведених тем.

## VNS practice work task №1

**Завдання 1.** Розробити лінійний алгоритм для розв'язання задачі.

**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.

```

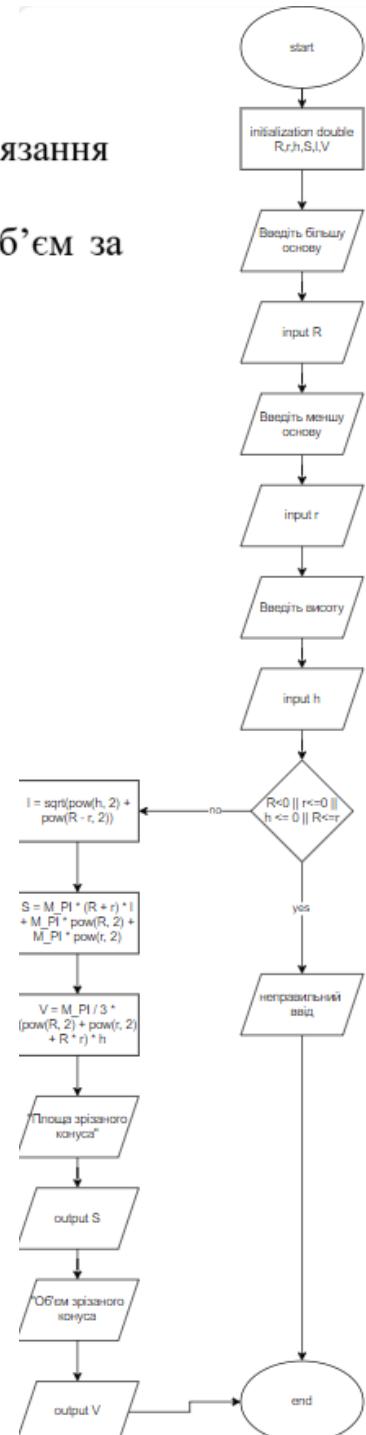
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double R, r, h, S, l, V;
8
9
10     cout << "Введіть більшу основу: ";
11     cin >> R;
12
13     cout << "Введіть меншу основу: ";
14     cin >> r;
15
16     cout << "Введіть висоту: ";
17     cin >> h;
18
19     if (R <= 0 || r <= 0 || h <= 0 || R <= r) {
20         cout << "неправильний ввід" << endl;
21         return 1;
22     }
23
24     l = sqrt(pow(h, 2) + pow(R - r, 2));
25     S = M_PI * (R + r) * l + M_PI * pow(R, 2) + M_PI * pow(r, 2);
26     V = M_PI / 3 * (pow(R, 2) + pow(r, 2) + R * r) * h;
27
28     cout << "Площа поверхні зрізаного конуса: " << S << endl;
29     cout << "Об'єм зрізаного конуса: " << V << endl;
30
31     return 0;
32 }

```

```

Введіть більшу основу: 6.25
Введіть меншу основу: 4.5
Введіть висоту: 7
Площа поверхні зрізаного конуса: 430.016
Об'єм зрізаного конуса: 640.95

```



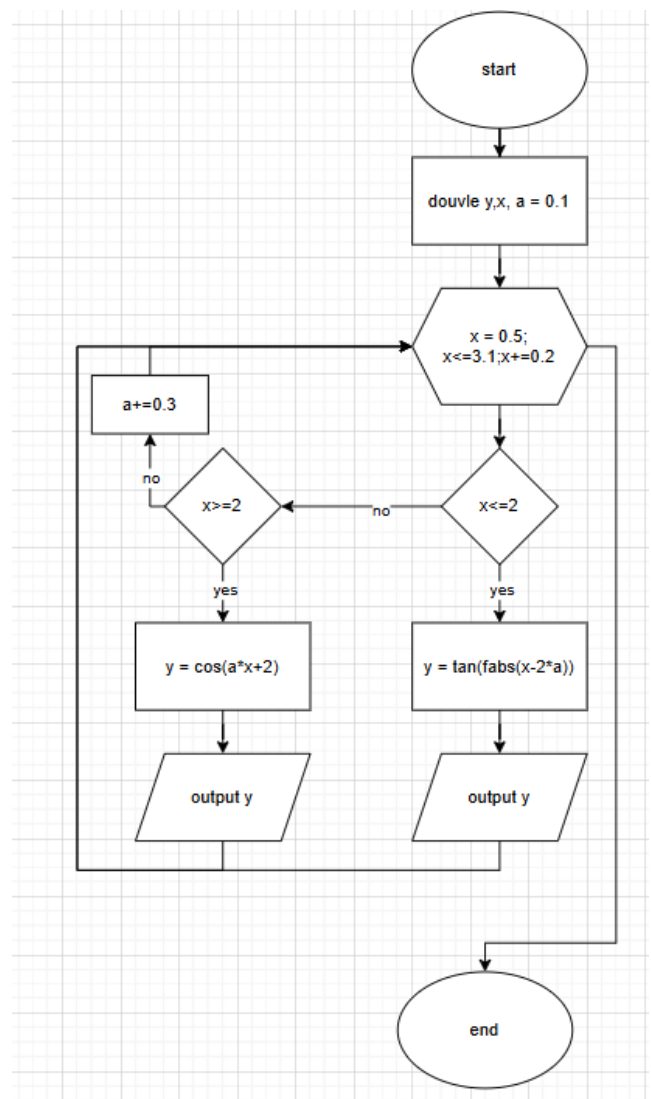
## VNS practice work task №2

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

**Варіант 16.** Обчислити функцію  $y = \begin{cases} \cos(ax+2), & x > 2, \\ \operatorname{tg}|x-2a|, & x \leq 2; \end{cases}$  де

$x \in [0.5; 3.1]$ ;  $h_x = 0.2$ ,  $a$  – має початкове значення 0.1 і змінюється одночасно зі змінною  $x$  з кроком  $h_a = 0.3$ .

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6
7  int main(){
8
9  double y,x, a = 0.1;
10
11  for( x = 0.5; x<=3.1; x+=0.2){
12
13  if(x<=2){
14      y = tan(fabs(x-2*a));
15      cout <<"y = " << y << endl;
16  }
17  if(x>2){
18      y = cos(a*x+2);
19      cout <<"y = " << y << endl;
20  }
21
22  a+=0.3;
23  }
24
25
26
27
28
29  return 0;
30 }
```



```
y = 0.309336  
y = 0.100335  
y = 0.546302  
y = 1.26016  
y = 3.6021  
y = -7.6966  
y = -1.70985  
y = -0.747022  
y = 0.567924  
y = -0.553048  
y = -0.94758  
y = 0.183382  
y = 0.986643
```

### VNS practice work task №3

**Завдання 3.** Написати програму згідно свого варіанту.

**Варіант 22.** Обчислення вартості покупки складається з декількох зошитів і такої ж кількості обкладинок до них. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення вартості покупки.

Введіть початкові дані:

Ціна зошита (грн.) > **2.75**

Ціна обкладинок (грн.) > **0.5**

Кількість комплектів (грн.) > **7**

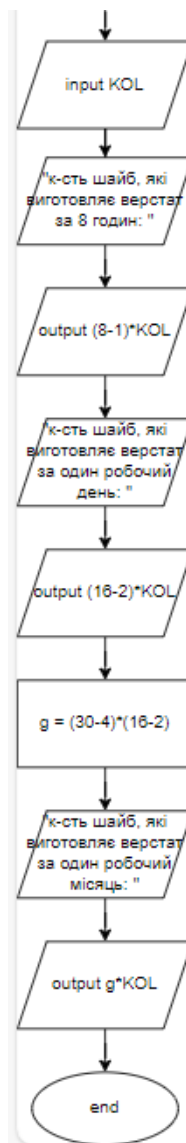
Вартість покупки: 15.45 грн.

Реалізувати виконання розрахунків: Верстат –автомат виготовляє в 1 годину КОЛ шайб. Скільки шайб він виготовить за 8-і годинну зміну, за робочий день при двозмінній роботі і за 30-ти денний місяць, якщо в місяці 4 вихідних дня і в кожній зміні верстат знаходиться 1 годину на профілактиці?

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7
8      float a,b,r;
9      int k;
10
11      cout << "Ціна зошита(грн): ";
12      cin >> a;
13      cout << "Ціна обкладинки(грн): ";
14      cin >> b;
15      cout << "К-сть комплектів: ";
16      cin >> k;
17
18      r = (a+b)*k;
19
20      cout << "Вартість покупки: " << r << endl;
21
22      int KOL,g;
23      cout << "к-сть шайб, які виготовляє верстат за годину: ";
24      cin >> KOL;
25
26      cout << "к-сть шайб, які виготовляє верстат за 8 годин: " << (8-1)*KOL << endl; //8-1 бо одна година профілактики
27      cout << "к-сть шайб, які виготовляє верстат за один робочий день: " << (16-2)*KOL << endl; //за 2 зміни 2 години профілактики
28      g = (30-4)*(16-2); // 26 днів він працює(має 4 вихідних)
29      //годину на профілактиці кожної зміни, тому за робочий день верстат працює 16-2= 14 год
30
31      cout << "к-сть шайб, які виготовляє верстат за один робочий місяць: " << g*KOL << endl; //16 годин - 2 зміни, проте за кожну зміну він стоїть
32
33
34
35      return 0;
36  }

```



```
Ціна зошита(грн): 5.25
Ціна обкладинки(грн): 1.22
К-сть комплектів: 6
Вартість покупки: 38.82
к-сть шайб, які виготовляє верстат за годину: 10
к-сть шайб, які виготовляє верстат за 8 годин: 70
к-сть шайб, які виготовляє верстат за один робочий день: 140
к-сть шайб, які виготовляє верстат за один робочий місяць: 3640
```

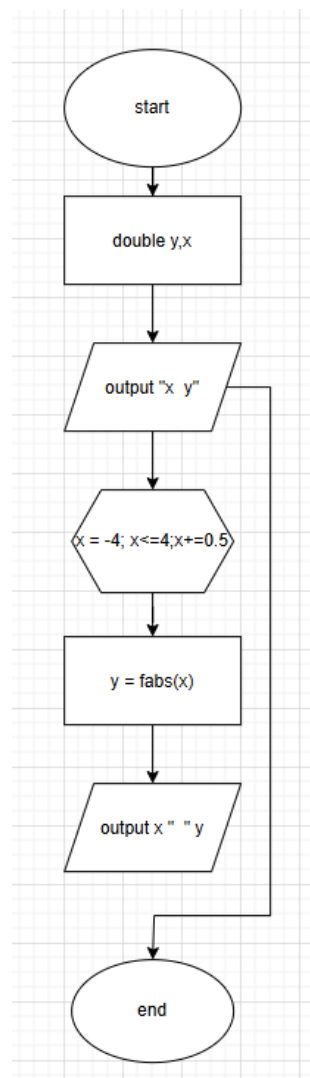
## VNS practice work task №4

**Завдання 4.** Написати програму згідно свого варіанту.

**Варіант 19.** Скласти програму, яка виводить таблицю значень функції  $y=|x|$ . Діапазон зміни аргументу -4 до 4, крок приросту аргументу 0,5.

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double y, x;
8
9      cout << "x\ty\n";
10
11     for (x = -4; x <= 4; x += 0.5) {
12         y = fabs(x);
13         cout << x << "\t" << y << endl;
14     }
15
16     return 0;
17 }
```

| x    | y   |
|------|-----|
| -4   | 4   |
| -3.5 | 3.5 |
| -3   | 3   |
| -2.5 | 2.5 |
| -2   | 2   |
| -1.5 | 1.5 |
| -1   | 1   |
| -0.5 | 0.5 |
| 0    | 0   |
| 0.5  | 0.5 |
| 1    | 1   |
| 1.5  | 1.5 |
| 2    | 2   |
| -1   | 1   |
| -0.5 | 0.5 |
| 0    | 0   |
| 0.5  | 0.5 |
| 1    | 1   |
| 1.5  | 1.5 |
| -1   | 1   |
| -0.5 | 0.5 |
| 0    | 0   |
| 0.5  | 0.5 |
| -1   | 1   |
| -0.5 | 0.5 |
| -1   | 1   |
| -1   | 1   |
| -1   | 1   |
| -1   | 1   |
| -1   | 1   |
| -0.5 | 0.5 |
| 0    | 0   |
| 0.5  | 0.5 |
| 1    | 1   |
| 1.5  | 1.5 |
| 2    | 2   |
| 2.5  | 2.5 |
| 3    | 3   |
| 3.5  | 3.5 |
| 4    | 4   |



## Task 5 - Algotester problem № 0183

## Breath of Fire

Limits: 2 sec., 256 MB

Senior told Dragon that last night he saw a fire over a forest **K** times from his house. Dragon remembers **N** locations where he has breathed fire over the forest yesterday. In order to be able to see a fire Senior has to be not further than **R** meters away from Dragon breathing fire.

You are given the location of Senior's house and fire locations Dragon remembers. Your task is to find the smallest valid value of **R**, i.e. Senior could saw a fire at least **K** times.

### Input

The first line contains two integers **X** and **Y** — coordinates of Senior's house. The second line contains two integers **N** and **K** separated by a single space. Each of the following **N** lines contains two integers **x<sub>i</sub>** and **y<sub>i</sub>** — coordinates of the **i**-th location where Dragon was breathing fire.

### Output

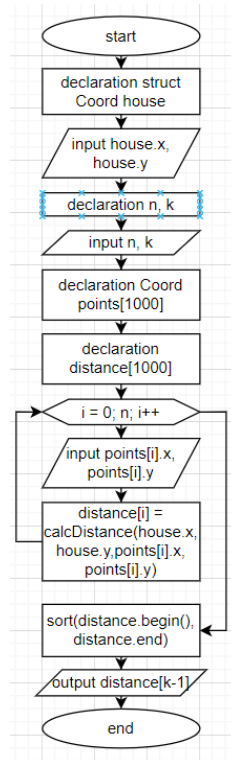
Print minimal valid value of **R**. An output will be considered as correct if the absolute or relative error does not exceed  $10^{-4}$ .

|             |                       |        |          |       |       |         |
|-------------|-----------------------|--------|----------|-------|-------|---------|
| an hour ago | 0183 - Breath of Fire | C++ 23 | Accepted | 0.003 | 1.066 | 1909716 |
|-------------|-----------------------|--------|----------|-------|-------|---------|

```
1  #include <iostream>
2  #include <cmath>
3  #include <algorithm>
4  #include <iomanip>
5  #include <vector>
6  using namespace std;
7
8  //0183
9  //breath of fire
10
11 struct Coord {
12     int x;
13     int y;
14 };
15
16 double calcDistance(int x1, int y1, int x2, int y2) {
17     return sqrt(static_cast<double>(pow(x1 - x2, 2) + pow(y1 - y2, 2)));
18 }
19
20 int main() {
21     Coord house;
22     cin >> house.x >> house.y;
23
24     int n, k;
25     cin >> n >> k;
26
27     vector<Coord> points(n);
28     vector<double> distance(n);
29
30     for (int i = 0; i < n; i++) {
31         cin >> points[i].x >> points[i].y;
32         distance[i] = calcDistance(house.x, house.y, points[i].x, points[i].y);
33     }
34
35     sort(distance.begin(), distance.end());
36     cout << setprecision(13) << distance[k - 1];
37     return 0;
38 }
```

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





## Task 6 - Algotester problem №0022

### 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.

#### Input

У першому рядку задано одне ціле число  $n$  — кількість друзів Зеника та Марічки.

У наступних  $n$  рядках задано по три цілі числа  $a_i$ ,  $b_i$ ,  $c_i$  — кількість борщу, картоплі та салату, яку хоче отримати  $i$ -ний друг.

#### Output

У єдиному рядку виведіть одне ціле число — сумарний ступінь недовіри всіх друзів до Зеника та Марічки.

21 minutes ago

0012 - Борщ, картопля і салат

C++ 23

Accepted

0.101

2.473 1909850

```

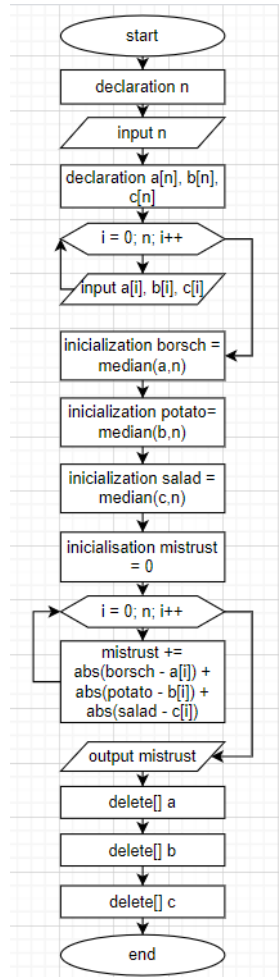
1  #include <iostream>
2  #include <algorithm>
3  #include <cmath>
4  using namespace std;
5
6  int median(int arr[], int n) {
7      sort(arr, arr + n);
8      return arr[n / 2];
9  }
10
11 int main() {
12     int n;
13     cin >> n;
14
15     int* a = new int[n];
16     int* b = new int[n];
17     int* c = new int[n];
18
19     for (int i = 0; i < n; i++) {
20         cin >> a[i] >> b[i] >> c[i];
21     }
22
23     int borsch = median(a, n);
24     int potato = median(b, n);
25     int salad = median(c, n);
26
27     long long mistrust = 0;
28     for (int i = 0; i < n; i++) {
29         mistrust += abs(borsch - a[i]) + abs(potato - b[i]) + abs(salad - c[i]);
30     }
31
32     cout << mistrust << endl;
33
34     delete[] a;
35     delete[] b;
36     delete[] c;
37
38     return 0;
39 }

```

```

2
10 50 100
70 10 50
150

```



## Task – 7 Algotester problem №0213

### Дуби та берези

Limits: 2 sec., 256 MiB

Усі ви знаєте, що студентам часто доручають надзвичайно важливу роботу, яку просто нікому виконати. Оце приборіть, 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 щойно відсвяткували День програміста, з радістю погодилися 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.

#### Input

У першому рядку задано цілі числа  $n$  і  $m$  — кількість студентів та кількість видів дерев відповідно.

У наступних  $n$  рядках задано результати, котрі записав один із студентів.

Результати, записані кожним із студентів —  $m$  цілих невід'ємних чисел,  $i$ -е з яких відповідає кількості дерев  $i$ -го виду, які знайшов у парку даний студент.

#### Output

У єдиному рядку виведіть одне ціле число — мінімальну можливу кількість дерев у парку.

|              |                       |        |          |       |               |
|--------------|-----------------------|--------|----------|-------|---------------|
| a minute ago | 0213 - Дуби та берези | C++ 23 | Accepted | 0.003 | 1.246 1909872 |
|--------------|-----------------------|--------|----------|-------|---------------|

```

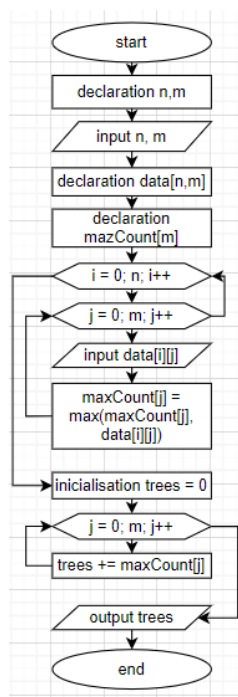
1  #include <iostream>
2  #include <algorithm>
3  #include <cmath>
4  #include <vector>
5
6  using namespace std;
7
8  int main()
9  {
10     int n, m;
11     cin >> n >> m;
12
13     vector<vector<int>> data(n, vector<int>(m));
14     vector<int> maxCount(m, 0);
15
16     for (int i = 0; i < n; i++)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

```

```

2 2
1 2
0 1
3

```



# Task – 8 Algotester problem №1051

## 47-й день року

Обмеження: 2 сек., 256 МБ

Шановні учасники олімпіади!

Команда Алготестера вітає усіх вас з великим святом — 47-м днем року! У цей день бажаємо вам хороших результатів на олімпіаді, успіху, ну і просто щастя.

За дивним збігом обставин, цього року обласна олімпіада припала саме на такий визначний день. Тож пропонуємо вам приєднатися до святкування, розв'язавши 7 нескладних задач про 47-й день року.

Перше завдання для вас на сьогодні — навчитися визначати, коли ж буде свято 47-го дня року в інших роки.

За заданим роком визначте дату 47-го дня цього року.

### Вхідні дані

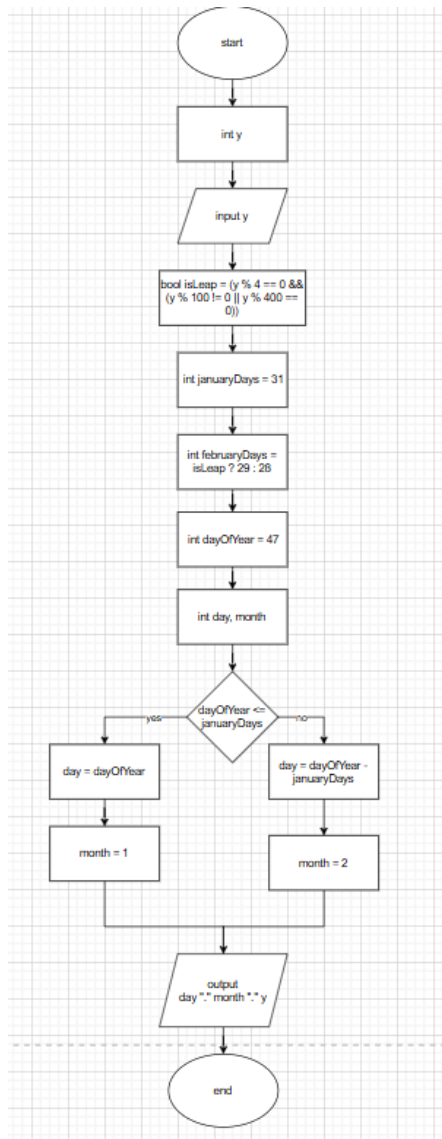
У єдиному рядку задано одне ціле число  $y$  — номер року.

### Вихідні дані

У єдиному рядку виведіть дату 47-го дня в заданому році у форматі `dd.mm.yyyy`.

|               |        |            |       |       |                          |
|---------------|--------|------------|-------|-------|--------------------------|
| 5 хвилин тому | C++ 23 | Зараховано | 0.002 | 1.191 | <a href="#">Перегляд</a> |
|---------------|--------|------------|-------|-------|--------------------------|

```
1  #include <iostream>
2  #include <iomanip>
3
4  using namespace std;
5
6  int main() {
7      int y;
8      cin >> y;
9
10     bool isLeap = (y % 4 == 0 && (y % 100 != 0 || y % 400 == 0));
11
12     int januaryDays = 31;
13     int februaryDays = isLeap ? 29 : 28;
14
15     int dayOfYear = 47;
16     int day, month;
17
18     if (dayOfYear <= januaryDays) {
19         day = dayOfYear;
20         month = 1;
21     } else {
22         day = dayOfYear - januaryDays;
23         month = 2;
24     }
25
26     cout << setw(2) << setfill('0') << day << "."
27          << setw(2) << setfill('0') << month << "." << y << endl;
28
29     return 0;
30 }
```



k\_8\_task\_bohdan\_s  
1991  
16.02.1991  
D5-G:\Hrany\Hrany\

## Висновок

**В результаті виконання цієї саги я переконався, що вмію та знаю як використовувати цикли, умовні оператори, функції вводу та виводу, створювати та ініціалізувати змінні та константи, використовувати структури, створювати та викликати функції, виконувати унарні та бінарні операції, працювати з циклами.**