

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-13
Кобзар Артем Сергійович

Тема: Розрахункова робота.

Мета роботи: Одержати практичні навички та закріпити знання, одержані протягом семестру.

Виконання роботи

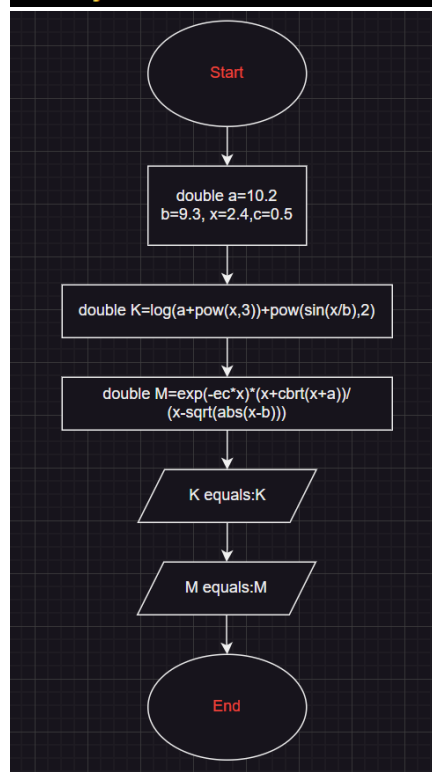
Завдання 1: VNS Practice Work Task 1

Розробити лінійний алгоритм для розв'язання задачі.

Варіант 18. $K = \ln(a + x^3) + \sin^2(x/b)$; $M = e^{-cx} \frac{x + \sqrt[3]{x+a}}{x - \sqrt{|x-b|}}$, де

$a=10,2$; $b=9,3$; $x=2,4$; $c=0,5$.

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4
5  int main() {
6      double a = 10.2, b = 9.3, x = 2.4, c = 0.5;
7      double K = log(a + pow(x, 3)) + pow(sin(x / b), 2);
8      double M = exp(-c * x) * (x + cbrt(x + a)) / (x - sqrt(abs(x - b)));
9      cout << "K equals: " << K << endl;
10     cout << "M equals: " << M << endl;
11     return 0;
12 }
```



K equals: 3.24419
M equals: -6.2779

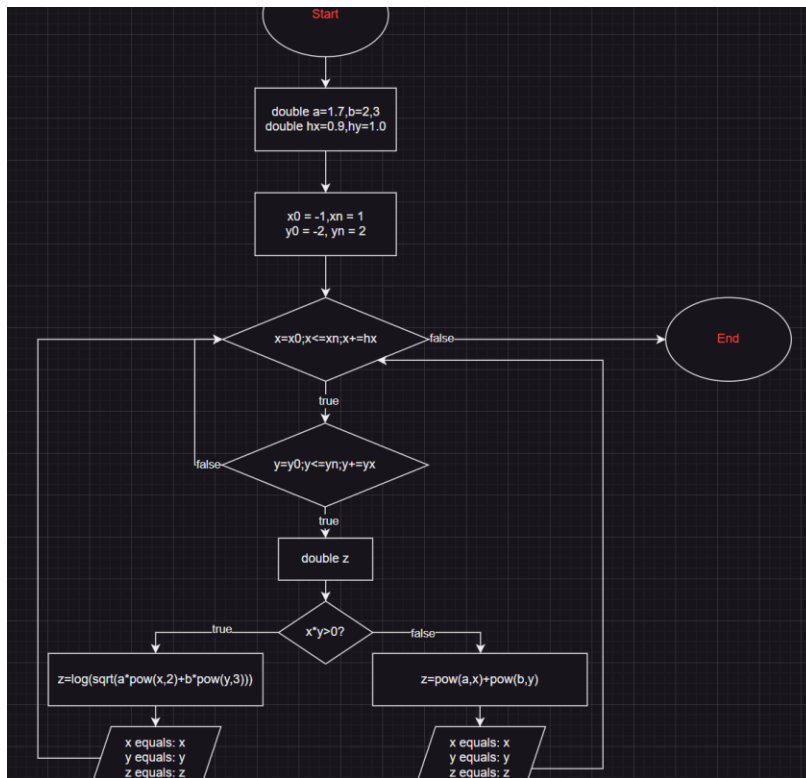
Завдання 2: VNS Practice Work Task 2

Розробити алгоритм, що розгалужується для розв'язання задачі.

$$\text{Варіант 13. } z = \begin{cases} \ln \sqrt{ax^2 + by^3}, & xy > 0, \\ a^x + b^y, & xy \leq 0; \end{cases} \text{ де } x \in [-1, 1]; h_x = 0,9;$$

$a = 1,7; b = 2,3; y \in [-2, 2]; h_y = 1.$

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double a = 1.7, b = 2.3;
8      double hx = 0.9, hy = 1.0;
9      double x0 = -1, xn = 1;
10     double y0 = -2, yn = 2;
11
12     for (double x = x0; x <= xn; x += hx) {
13         for (double y = y0; y <= yn; y += hy) {
14             double z;
15
16             if (x * y > 0) {
17                 z = log(sqrt(a * pow(x, 2) + b * pow(y, 3)));
18             } else {
19                 z = pow(a, x) + pow(b, y);
20             }
21
22             cout << "x equals: " << x << endl << "y equals: " << y << endl << "z equals: " << z << endl;
23         }
24     }
25     return 0;
26 }
```



```

x equals: -1
y equals: -2
z equals: nan
x equals: -1
y equals: -1
z equals: nan
x equals: -1
y equals: 0
z equals: 1.58824
x equals: -1
y equals: 1
z equals: 2.88824
x equals: -1
y equals: 2
z equals: 5.87824
x equals: -0.1
y equals: -2
z equals: nan
x equals: -0.1
y equals: -1
z equals: nan
x equals: -0.1
y equals: 0
z equals: 1.94832
x equals: -0.1
y equals: 1
z equals: 3.24832
x equals: -0.1
y equals: 2
z equals: 6.23832
x equals: 0.8
y equals: -2
z equals: 1.71787
x equals: 0.8
y equals: -1
z equals: 1.96361
x equals: 0.8
y equals: 0
z equals: 2.52883
x equals: 0.8
y equals: 1
z equals: 0.61012

```

Завдання 3: VNS Practice Work Task 3

Варіант 13. Обчислення об'єму циліндра. Нижче приведений вид екрану під час виконання програми, що рекомендується (дані, які вводяться користувачем, виділені напівжирним шрифтом).

Обчислення об'єму циліндра.

Введіть початкові дані:

Радіус підстави (см) > **5.5**

Висот циліндра (см) > **7**

Об'єм 665.23 см.куб.

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      double r, h, V;
8      cout<<"Обчислення об'єму циліндра."<<endl;
9      cout<<"Введіть початкові дані: ";
10     cin>>r>>h;
11     cout<<endl<<"Радіус підставки(см): "<<r;
12     cout<<endl<<"Висота циліндра(см): "<<h;
13     V = M_PI*pow(r,2)*h;
14     cout<<endl<<"Об'єм: " <<V<<" см. куб. "<<endl;
15 }

```



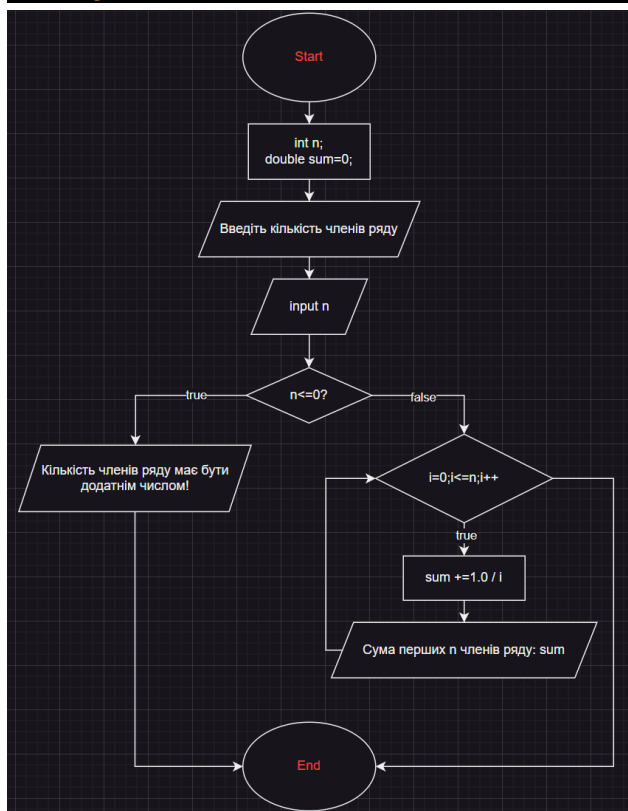
Обчислення об'єму циліндра.
Введіть початкові дані: 6 8

Радіус підставки(см): 6
Висота циліндра(см): 8
Об'єм: 904.779 см. куб.

Завдання 4: VNS Practice Work Task 4

Варіант 8. Написати програму, яка обчислює суму перших n членів ряду: $1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \dots$

```
1  #include <iostream>
2  using namespace std;
3
4  int main() {
5      int n;
6      double sum = 0;
7
8      cout << "Введіть кількість членів ряду: ";
9      cin >> n;
10
11     if (n <= 0) {
12         cout << "Кількість членів ряду має бути додатнім числом!" << endl;
13         return 1;
14     }
15
16     for (int i = 1; i <= n; i++) {
17         sum += 1.0 / i;
18     }
19     cout << "Сума перших " << n << " членів ряду: " << sum << endl;
20
21     return 0;
22 }
```



Введіть кількість членів ряду: 5
Сума перших 5 членів ряду: 2.28333

Завдання 5: Algotester Practice Task 1

Марічка і печиво

Обмеження: 2 сек., 256 МБ

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 підлідила, 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.


```

#include <iostream>

using namespace std;

int main (){

    int n;
    cin >> n;
    long long result=0;
    long long cookies;

    for ( int i = 0 ; i < n ; i++){
        cin >> cookies;
        result += cookies - 1;
    }
    cout << result;
    return 0;
}

```

```

4
4 7 47 74
128

```

2 дні тому

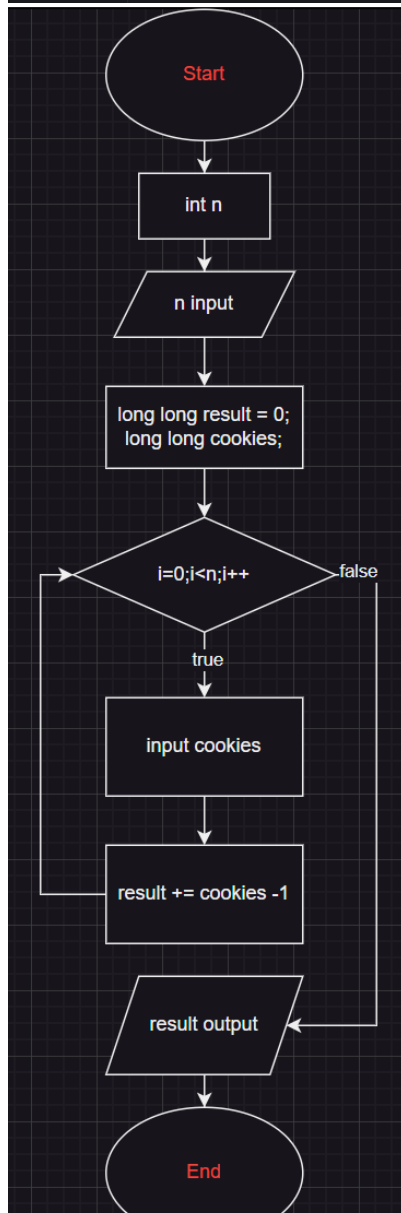
0011 - Марічка і печиво

C++ 23

Зараховано

0.037

1.188 1906090



Завдання 6: Algotester Practice Task 2

Апельсини

Обмеження: 2 сек., 256 MiB

Діти міряються різними речима. У нашій задачі — апельсинами. Марічка і Софійка міряються з Петриком.

Потрібно визначити, чи Марічка й Софійка разом мають більше апельсинів, ніж Петрик.

Вхідні дані

У першому і єдиному рядку задано три цілі числа a , b та c — кількість апельсинів у Марічки, Софійки і Петрика відповідно.

Вихідні дані

У єдиному рядку виведіть відповідь до задачі — **YES**, якщо дівчата разом мають більше апельсинів, ніж Петрик, або **NO** в іншому випадку.

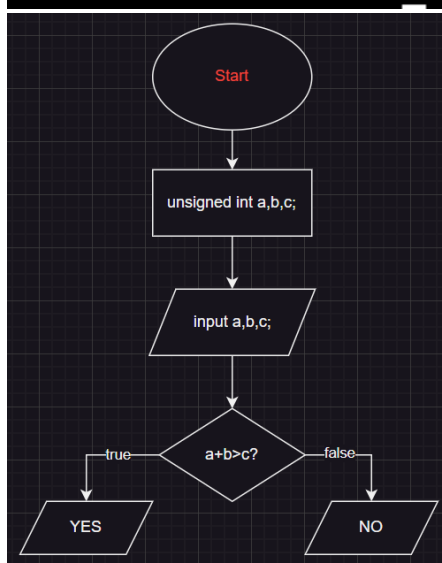
```
1  #include <iostream>
2
3  int main() {
4      unsigned int a,b,c;
5      std::cin>>a>>b>>c;
6      if(a+b>c) {
7          std::cout<<"YES";
8      } else {
9          std::cout<<"NO";
10     }
11     return 0;
12 }
```

3 5 7

YES

12 10 25

NO



Завдання 7: Algotester Practice Task 3

Дуби та берези

Обмеження: 2 сек., 256 MiB

Усі ви знаєте, що студентам часто доручають надзвичайно важливу роботу, яку просто нікому виконати. Оце приберіть, оце зберіть, туди сходите, 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 молоді 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.

Вхідні дані

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

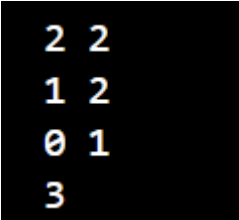
У наступних n рядках задано результати, котрі записав один із студентів.

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.

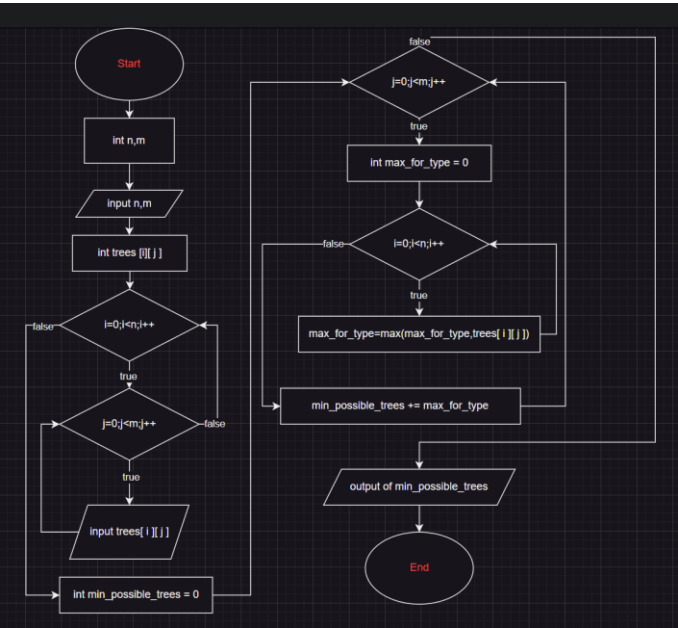
Вихідні дані

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

```
1 #include <iostream>
2 #include <algorithm>
3
4 int main() {
5     int n, m;
6     std::cin >> n >> m;
7
8     int trees[n][m];
9
10    for (int i = 0; i < n; i++) {
11        for (int j = 0; j < m; j++) {
12            | std::cin >> trees[i][j];
13        }
14    }
15    int min_possible_trees = 0;
16
17    for (int j = 0; j < m; j++) {
18        int max_for_type = 0;
19        for (int i = 0; i < n; i++) {
20            | max_for_type = std::max(max_for_type, trees[i][j]);
21        }
22        min_possible_trees += max_for_type;
23    }
24
25    std::cout << min_possible_trees << std::endl;
26
27    return 0;
28 }
```



2 дні тому	0213 - Дуби та берези	C++ 23	Зараховано	0.004	1.188	1906156
------------	-----------------------	--------	------------	-------	-------	---------



Завдання 8: Algotester Practice Task 4

Щасливий результат

Обмеження: 2 сек., 256 МБ

Після завершення основного туру олімпіади з програмування Зеник отримав невеличкий клптик паперу, на якому було надруковане число x — кількість балів, що набрав Зеник. Зауважте, що згідно з кращими традиціями олімпіади з програмування, кількість балів Зеника не може бути нульовою чи від'ємною.

Помітивши не дуже щасливе обличчя Зеника, Марічка нагадала йому про щасливі цифри. Як ви вже напевно знаєте, щасливими вважають цифри 4 та 7. Марічка запевнила Зеника, що найкращим є не найбільший результат, а той, десятковий запис якого містить найбільше щасливих цифр.

Вам необхідно допомогти юному учаснику олімпіади з програмування та порахувати кількість щасливих цифр у його результаті.

Вхідні дані

У єдиному рядку задано одне ціле число x — результат Зеника.

Вихідні дані

У єдиному рядку виведіть одне ціле число — кількість щасливих цифр у десятковому записі x .

```

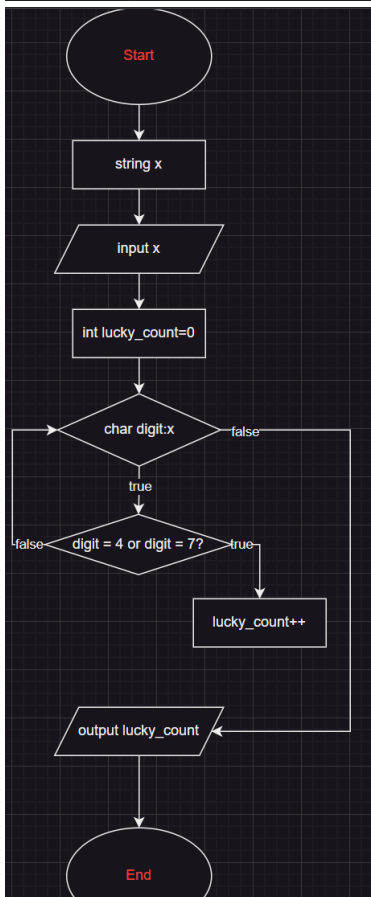
#include <iostream>
#include <string>
using namespace std;

int main() {
    string x;
    cin >> x;
    int lucky_count = 0;
    for (char digit : x) {
        if (digit == '4' || digit == '7') {
            lucky_count++;
        }
    }
    cout << lucky_count << endl;
    return 0;
}

```

4072457
4

2 дні тому 0532 - Щасливий результат C++ 23 Зараховано 0.002 1.074 1906364



Висновок: Під час виконання цієї розрахункової роботи, я одержав практичні навчки та закріпив знання, одержані протягом семестру.