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-13

Ходацька Аліна Віталіївна

Львів - 2024

Мета роботи:

Одержати практичні навички в розробці і дослідженні алгоритмів розв’язання задач.

Теоретичні відомості:

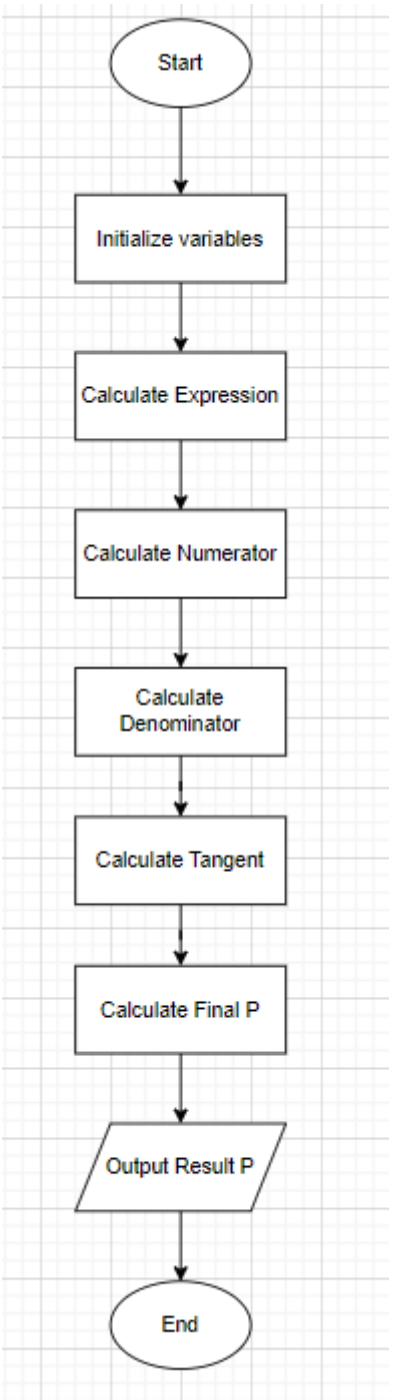
1. Теми, необхідні для виконання роботи:
    - Пройдені під час семестру.
2. Джерела використані для ознайомлення з вищезазначеними темами:
    - Ознайомлені під час навчання.

Виконання роботи

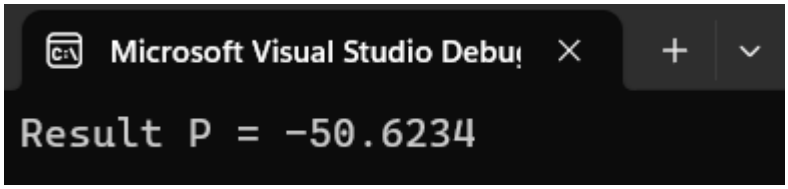
Завдання №1 VNS Task 1 Variant 12

Розробити лінійний алгоритм для розв’язання задачі.

Варіант 12. $P = \left| \frac{\sin^3(ax^3 + by^2 - ab)}{\sqrt[3]{(ax^3 + by^2 - a)^2 + \pi}} \right| + tg(ax^3 + by^2 - ab)$, де x=0,25; y=1,31; a=3,5; b=0,9.



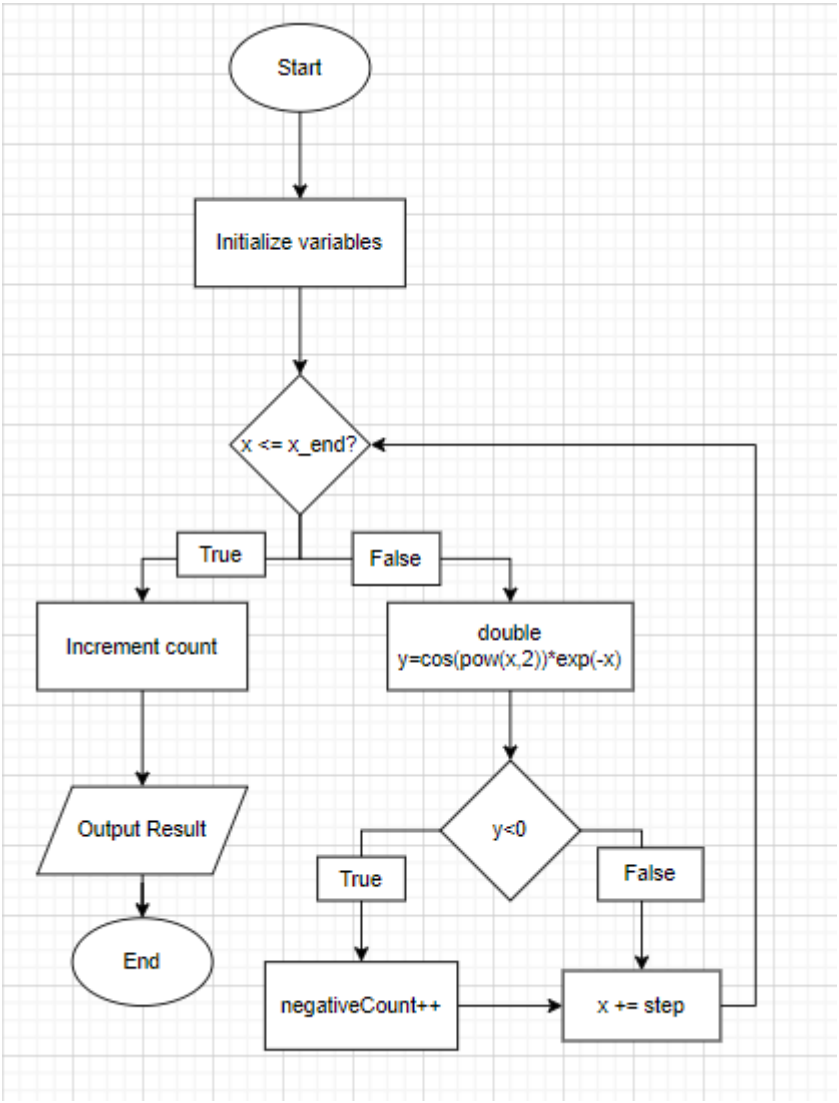
```cpp
#include <iostream>
#include <cmath>

#define M_PI 3.14159265358979323846

using namespace std;

int main() {
    double x = 0.25;
    double y = 1.31;
    double a = 3.5;
    double b = 0.9;

    double expression = a * pow(x, 3) + b * pow(y, 2) - a * b;
    double numerator = pow(sin(expression), 3);
    double denominator = cbrt(pow(expression, 2) + M_PI);  // cbrt - cube root
    double tangent = tan(expression);

    double P = numerator / denominator + tangent;

    cout << "Result P = " << P << endl;

    return 0;
}
```



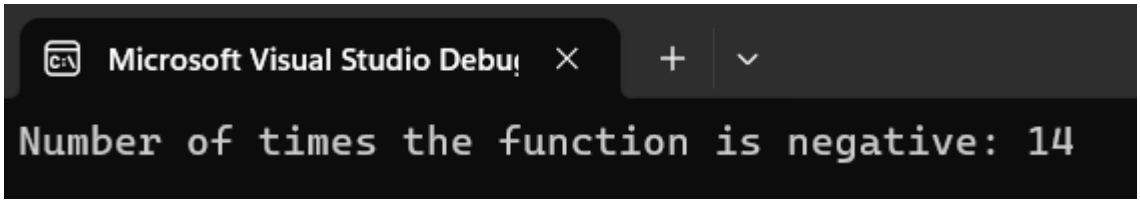
Завдання №2 VNS Task 2 Variant 19

Розробити алгоритм, що розгалужується для розв’язання задачі.

Варіант 19. Підрахувати, скільки разів функція $y = \cos x^2 \cdot e^{-x}$ приймає негативне значення, якщо $x \in [0{,}3;5]$; $h_x = 0{,}1$.



```cpp
#include <iostream>
#include <cmath>

using namespace std;

int main() {
    double x_start = 0.0;
    double x_end = 3.5;
    double step = 0.1;

    int negativeCount = 0;

    for (double x = x_start; x <= x_end; x += step) {
        double y = cos(pow(x, 2)) * exp(-x);

        if (y < 0) {
            negativeCount++;
        }
    }

    cout << "Number of times the function is negative: " << negativeCount << endl;

    return 0;
}
```



Завдання №3 VNS Task 3 Variant 7

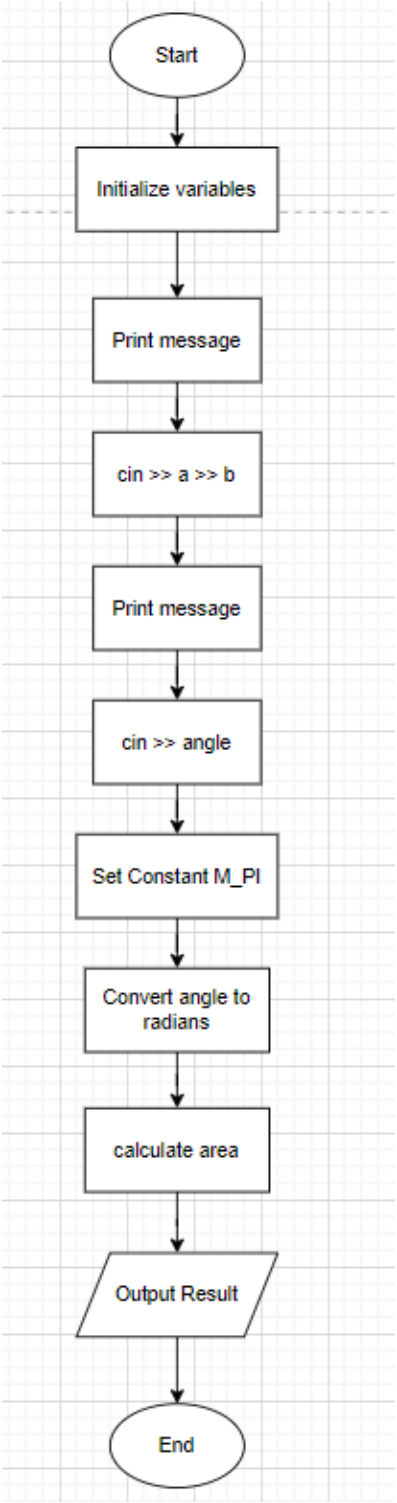
Обчислення площі трикутника, якщо відомі довжини двох його сторін і величина кута між цими сторонами.

Обчислення площі трикутника.

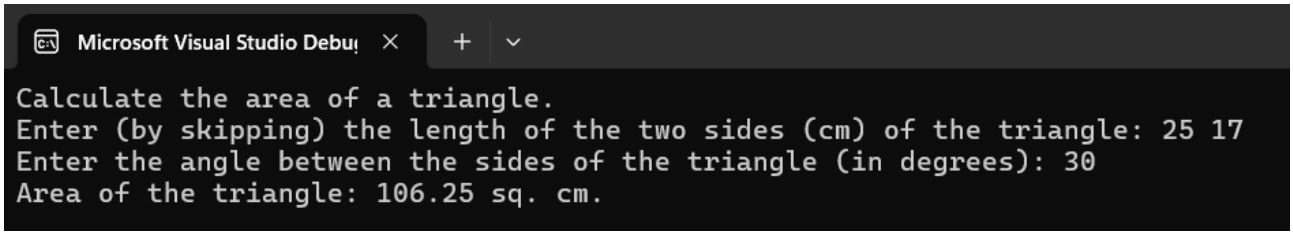
Введіть (через пропуск) довжини двох сторін (см) трикутника > *25 17*

Введіть величину кута між сторонами трикутника > *30*

Площа трикутника: 106.25 кв.см.



```cpp
#include <iostream>
#include <cmath>  // For using the sin function

using namespace std;

int main() {
    double a, b, angle, area;

    cout << "Calculate the area of a triangle." << endl;

    cout << "Enter (by skipping) the length of the two sides (cm) of the triangle: ";
    cin >> a >> b;

    cout << "Enter the angle between the sides of the triangle (in degrees): ";
    cin >> angle;

    const double M_PI = 3.14159265358979323846;

    // Convert the angle from degrees to radians
    double angle_rad = angle * M_PI / 180.0;

    area = 0.5 * a * b * sin(angle_rad);

    cout << "Area of the triangle: " << area << " sq. cm." << endl;

    return 0;
}
```



Завдання №4 VNS Task 4 Variant 14

Обчислення площі поверхні циліндра.

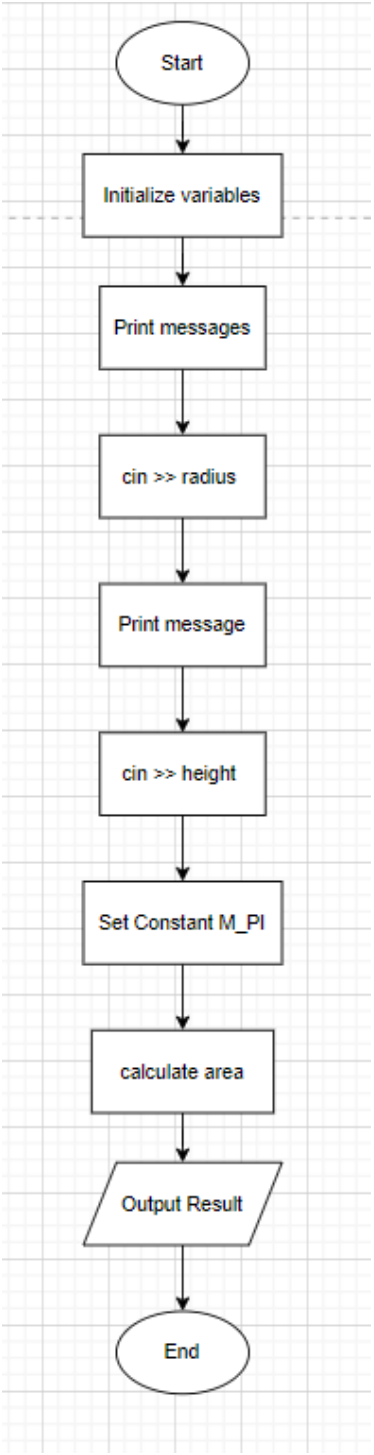
Введіть початкові дані:

Радіус підстави (см) > *5.5*

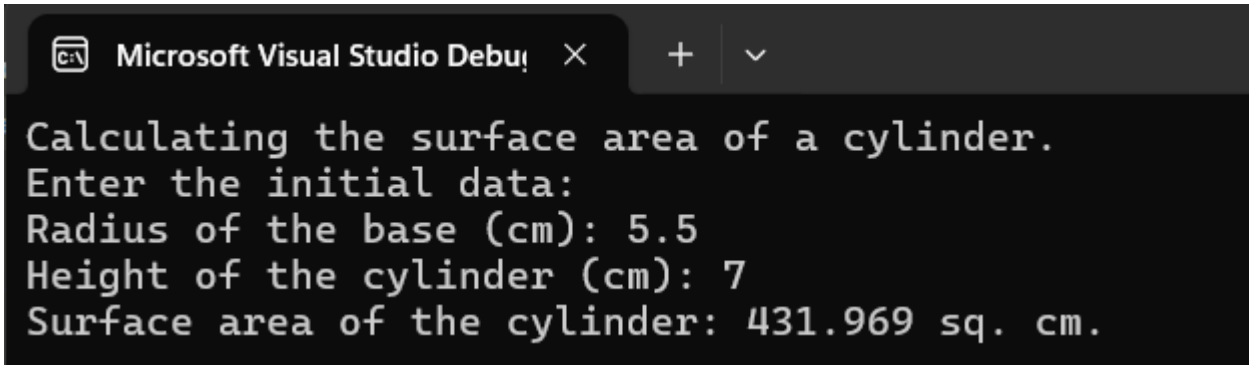
Висота циліндра (см) > *7*

Площа поверхні циліндра: 431.97 кв.см.

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



```cpp
#include <iostream>
#include <cmath>  // For using the M_PI constant and functions

using namespace std;

int main() {
    double radius, height, surface_area;

    cout << "Calculating the surface area of a cylinder." << endl;

    cout << "Enter the initial data:" << endl;
    cout << "Radius of the base (cm): ";
    cin >> radius;

    cout << "Height of the cylinder (cm): ";
    cin >> height;

    const double M_PI = 3.14159265358979323846;

    // Calculate the surface area of the cylinder using the formula
    surface_area = 2 * M_PI * radius * radius + 2 * M_PI * radius * height;

    cout << "Surface area of the cylinder: " << surface_area << " sq. cm." << endl;

    return 0;
}
```



Завдання №5 Algotester Task “Загадкове число”

Загадкове число

Limits: 2 sec., 256 MiB

Марічка, як і годиться усім представницям прекрасної статі, полюбляє говорити загадками. Так і цього разу, Марічка планувала написати Зенику записку з її улюбленим числом, проте вона не втрималася, і деякі цифри в числі замінила на зірочки (символ `*`).

Коли Зеник отримав записку, він, як завжди, не зрозумів, що це мало означати і який прихований зміст у цьому повідомленні. Тому він вирішив визначити, яке мінімальне, та яке максимальне можливе число Марічка хотіла йому сказати.

Зверніть увагу, кожну зірочку треба замінити на якусь цифру. Також, числа не повинні містити ведучих нулів. Тобто, першою цифрою числа повинен бути не 0.

Input

У єдиному рядку задано один рядок — послідовність з цифр і зірочок.

Output

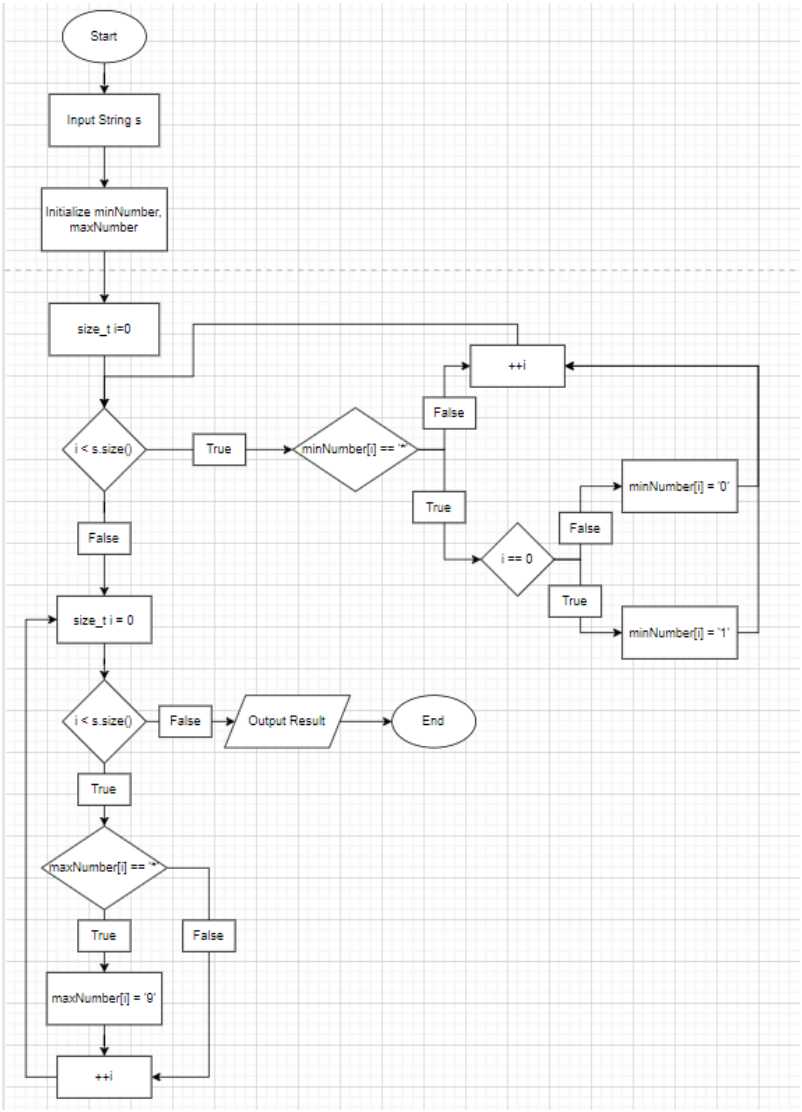
У єдиному рядку виведіть два натуральних числа — мінімальне та максимальне можливе число, яке Марічка хотіла повідомити.

Constraints

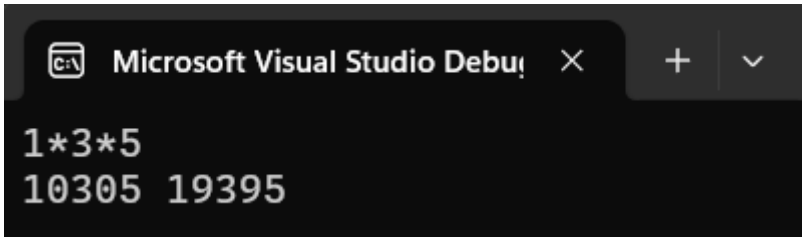
Довжина числа не перевищує 9,

гарантується, що перший символ не 0.

Реалізувати програму, яка за введеним рядком, що містить цифри та символи *, обчислює мінімальне та максимальне можливе натуральне число, замінюючи кожен * відповідно на найменшу або найбільшу можливу цифру. Результат вивести у вигляді двох чисел через пробіл.



```cpp
#include <iostream>
#include <string>

int main() {
    std::string s;
    std::cin >> s;

    std::string minNumber = s;
    std::string maxNumber = s;

    // Обчислення мінімального числа
    for (size_t i = 0; i < s.size(); ++i) {
        if (minNumber[i] == '*') {
            if (i == 0) {
                minNumber[i] = '1'; // Перший символ не може бути 0
            }
            else {
                minNumber[i] = '0';
            }
        }
    }

    // Обчислення максимального числа
    for (size_t i = 0; i < s.size(); ++i) {
        if (maxNumber[i] == '*') {
            maxNumber[i] = '9';
        }
    }

    // Виведення результатів
    std::cout << minNumber << " " << maxNumber << std::endl;

    return 0;
}
```



Завдання №6 Algotester Task “Музикант мобільний”

Музикант мобільний

Limits: 2 sec., 256 MiB

Навіть до Музиканта дійшла цивілізація! Уявляєте, Музикант та мобільний зв’язок — сміху гарантовано як мінімум на 20% більше. Музикант користується тарифним планом «Ретро-базар».

За підключення йому приходиться платити 11 гривень, кожна з перших 7 хвилин коштує 9 гривень, а кожна наступна — 5. Тарифікація похвилинна, тобто навіть якщо Музикант не використав повністю час останньої хвилини, він усе одно заплатить за неї повністю.

Вам необхідно знайти суму, витрачену Музикантом на телефонний дзвінок, який тривав n секунд.

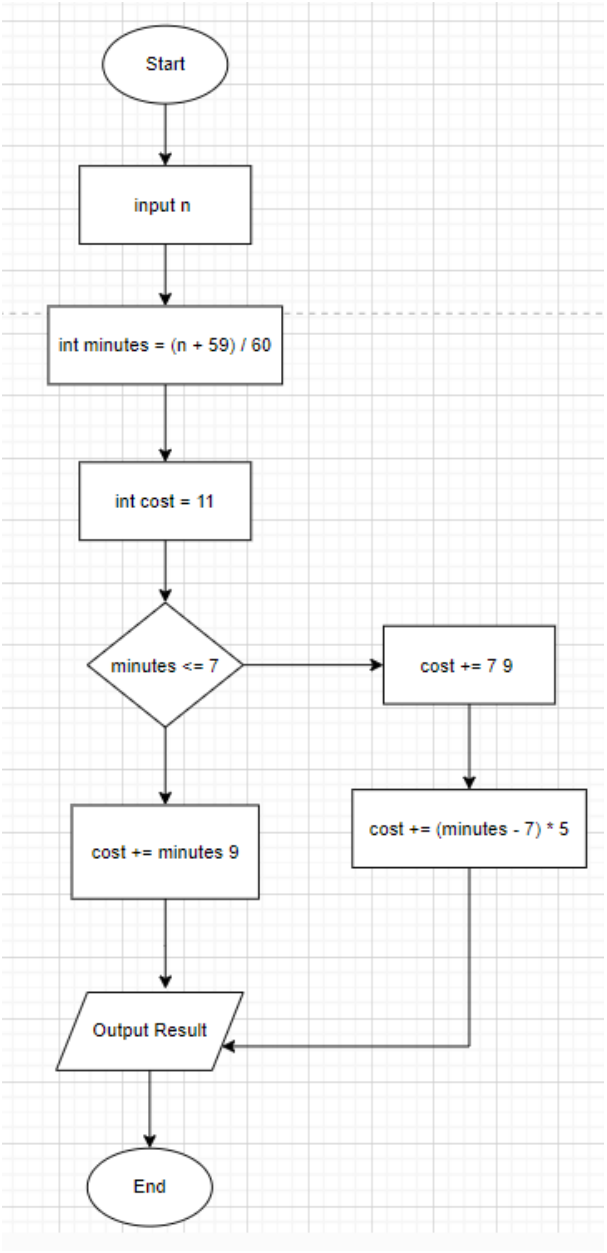
Input

Єдиний рядок містить ціле число n — кількість секунд.

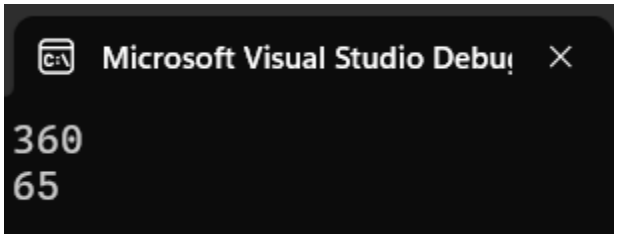
Output

У єдиному рядку виведіть ціле число — вартість дзвінка.

Реалізувати програму, яка за заданою кількістю секунд n обчислює вартість телефонного дзвінка за тарифним планом «Ретро-базар», враховуючи похвилинну тарифікацію та зміну вартості після перших 7 хвилин. Результат вивести як ціле число.



```cpp
#include <iostream>
#include <cmath>

int main() {
    int n;
    std::cin >> n;

    int minutes = (n + 59) / 60;

    int cost = 11;

    if (minutes <= 7) {
        cost += minutes * 9;
    }
    else {
        cost += 7 * 9;
        cost += (minutes - 7) * 5;
    }

    std::cout << cost << std::endl;

    return 0;
}
```



Завдання №7 Algotester Task “The Food”

The Food

Limits: 2 sec., 256 MiB

There are N plates with food. Each plate is described by its food name and its weight. You want to take some of them, but you don’t want to have two plates with the same food (i.e. with the same food name). Find the maximal possible total weight of the food you can take.

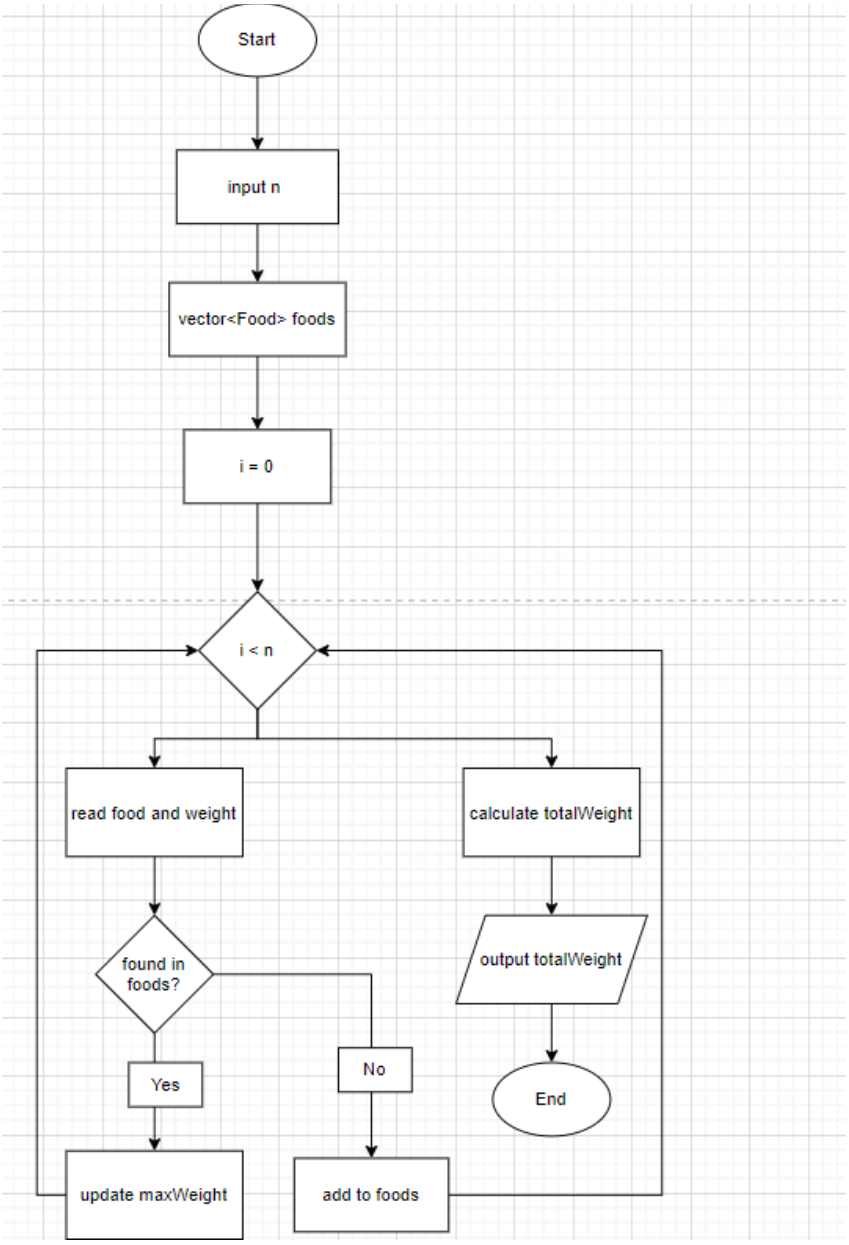
Input

The first line contains the integer N. Each of the following N lines contains a string S_j and an integer W_j separated with a single space. Here S_j is the name of the food on the j-th plate and W_j is the weight of the j-th plate.

Output

The only integer which is the maximal possible total food weight.

Реалізувати програму, яка за заданою кількістю тарілок з їжею N, де кожна тарілка має назву їжі та її вагу, обчислює максимальну сумарну вагу їжі, яку можна взяти, враховуючи, що тарілки з однаковою назвою їжі можна вибрати лише одну. Результат вивести у вигляді одного цілого числа.



```cpp
#include <iostream>
#include <string>
#include <vector>
using namespace std;

struct Food {
    string name;
    int maxWeight;
};

int main() {
    int N;
    cin >> N;

    vector<Food> foods;

    for (int i = 0; i < N; ++i) {
        string food;
        int weight;
        cin >> food >> weight;

        bool found = false;

        for (auto& f : foods) {
            if (f.name == food) {
                f.maxWeight = max(f.maxWeight, weight);
                found = true;
                break;
            }
        }

        if (!found) {
            foods.push_back({ food, weight });
        }
    }

    int totalWeight = 0;
    for (const auto& f : foods) {
        totalWeight += f.maxWeight;
    }

    cout << totalWeight << endl;

    return 0;
}
```

```
Microsoft Visual Studio Debu
7
potato 4
soup 2
potato 9
steak 2
steak 5
potato 4
rice 1
17
```

Завдання №8 Algotester Task “Аморальні експерименти”

Аморальні експерименти

Limits: 2 sec., 256 MiB

Щоб розробити ефективну вакцину від реп’яховірусу, потрібно розуміти, наскільки швидко він розповсюджується. Провідні українські вчені — Зеник та Марічка — проводять різноманітні експерименти, щоб краще розуміти закономірності розповсюдження вірусу.

Цього разу вони розсадили n людей за круглим столом. Для кожної людини відомо, чи вона заражена на початку експерименту. Зеник припускає, що якщо людина хвора, то рівно за одну хвилину вона заражає людей, що сидять поряд з нею за столом.

Якщо припущення Зеника правильне, то через скільки часу всі люди за столом будуть зараженими?

Input

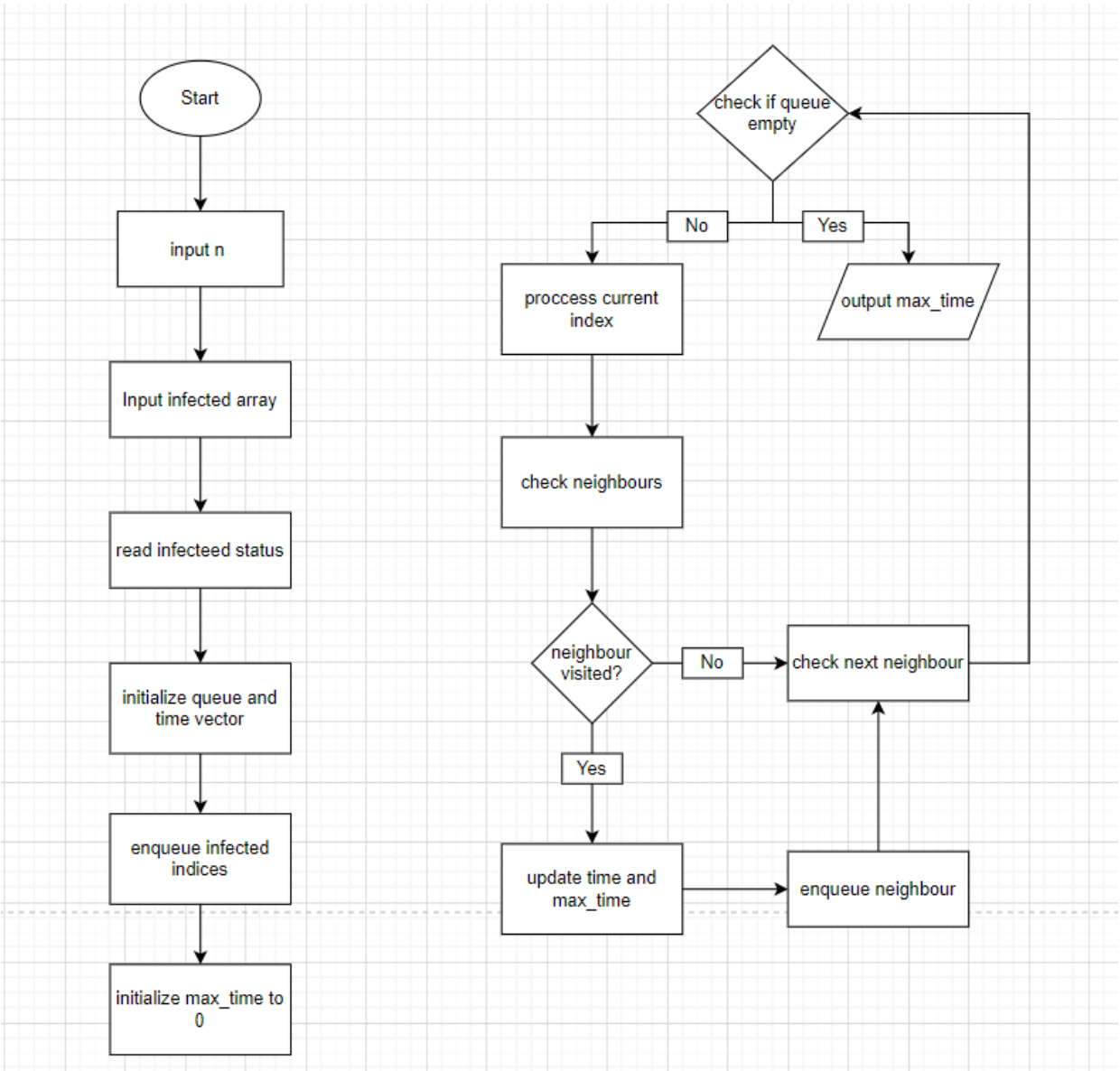
У першому рядку задано ціле число n — кількість людей, що беруть участь в експерименті.

У другому рядку задано n чисел 0 або 1. 1 означає, що відповідний учасник експерименту є хворим на початку експерименту, 0 — здоровим.

Output

У єдиному рядку виведіть ціле число — мінімальний час, через який усі учасники експерименту захворіють, якщо припущення Зеника є правильним.

Реалізувати програму, яка визначає мінімальний час, через який усі n людей, розсаджених за круглим столом, стануть зараженими, враховуючи, що хвора людина заражає сусідів за одну хвилину. На вхід подається кількість людей та їхній початковий стан (0 — здоровий, 1 — хворий).



```cpp
#include <iostream>
#include <vector>
#include <queue>

int main() {
    int n;
    std::cin >> n;

    std::vector<int> infected(n);
    for (int i = 0; i < n; ++i) {
        std::cin >> infected[i];
    }

    std::queue<int> q;
    std::vector<int> time(n, -1);

    for (int i = 0; i < n; ++i) {
        if (infected[i] == 1) {
            q.push(i);
            time[i] = 0;
        }
    }

    int max_time = 0;
    while (!q.empty()) {
        int current = q.front();
        q.pop();

        for (int i = -1; i <= 1; i += 2) {
            int neighbor = (current + i + n) % n;
            if (time[neighbor] == -1) {
                time[neighbor] = time[current] + 1;
                max_time = std::max(max_time, time[neighbor]);
                q.push(neighbor);
            }
        }
    }

    std::cout << max_time << std::endl;
    return 0;
}
```

Посилання на pull request: https://github.com/artificial-intelligence-department/ai_programming_playground_2024/pull/557