

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 ІІІ-12
Іванів Христина

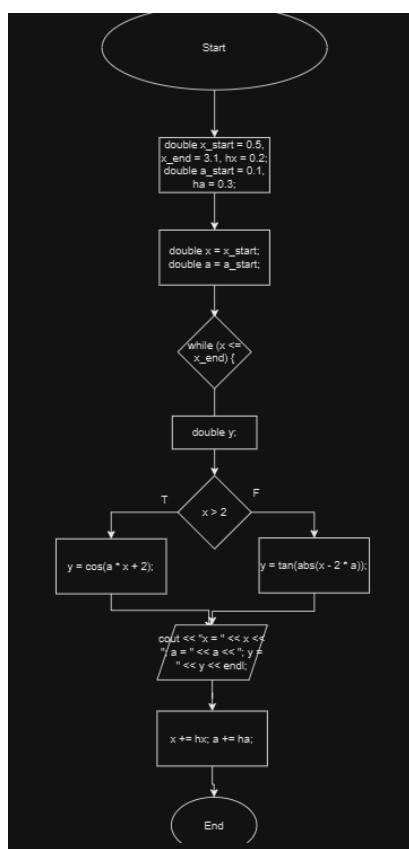
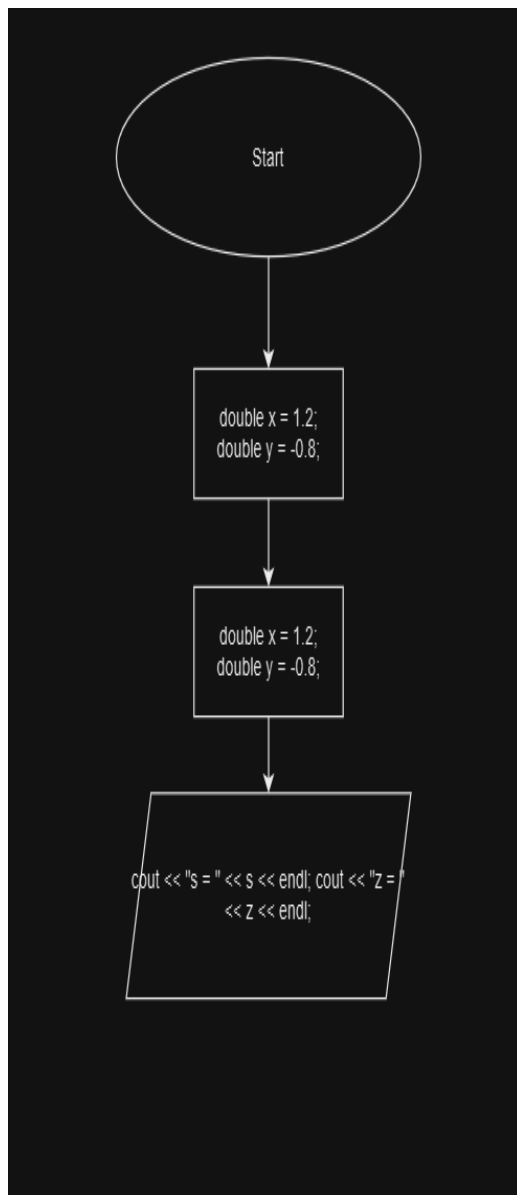
Мета роботи: одержати практичні навички в розробці і дослідженні алгоритмів розв'язання задач.

Завдання 1

Розробити лінійний алгоритм для розв'язання задачі:

Варіант 3. $s = 1 + x + \frac{x^2}{2} + \frac{x^3}{3} + \frac{x^4}{4}$; $z = (\sin x^3 + \cos^2 y)$, де $x=1,2$,
 $y=-0,8$.

```
1 #include <iostream>
2
3 using namespace std;
4
5 int main() {
6     int count;
7     cout << "Введіть кількість
8     cin >> count;
9
10    if (count <= 0) {
11        cout << "Кількість чисел
12        return 1;
13    }
14
15    double sum = 0.0, number;
16
17    cout << "Введіть числа:" << endl;
18    for (int i = 0; i < count; i++)
19        cout << "Число " << i + 1 << ": ";
20    cin >> number;
21    sum += number;
22
23
24    double average = sum / count;
25    cout << "Середнє арифметичне: " << average << endl;
26
27    return 0;
28 }
```



Завдання 2. Розробити алгоритм, що розгалужується для розв'язання задачі номер якої відповідає списку функцій $y = \begin{cases} \cos(ax + 2), & x > 2, \\ tg|x - 2a|, & x \leq 2; \end{cases}$ де a має початкове значення 0,1 і змінюється кроком $h_a = 0,3$.

порядковому номеру студента в журналі викладача

```

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

```

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**

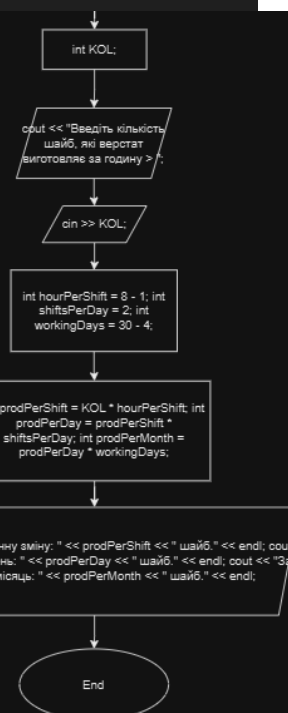
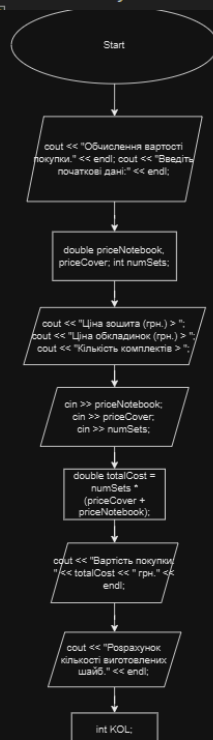
Вартість покупки: 15.45 грн.

Реалізувати виконання розрахунків: Верстат –автомат виготовляє в 1 годину KOL 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     // 1. обчислення вартості покупки
7     cout << "Обчислення вартості покупки." << endl;
8     cout << "Введіть початкові дані:" << endl;
9
10    double priceNotebook, priceCover;
11    int numSets;
12
13    cout << "Ціна зошита (грн.) > ";
14    cin >> priceNotebook;
15    cout << "Ціна обкладинок (грн.) > ";
16    cin >> priceCover;
17    cout << "Кількість комплектів > ";
18    cin >> numSets;
19
20    double totalCost = numSets * (priceCover + priceNotebook);
21
22    cout << "Вартість покупки: " << totalCost << " грн." << endl;
23
24
25    // 2. розрахунок кількості шайб
26    cout << "Розрахунок кількості виготовлених шайб." << endl;
27
28    int KOL;
29    cout << "Введіть кількість шайб, які верстат виготовляє за годину > ";
30    cin >> KOL;
31
32    int hourPerShift = 8 - 1;
33    int shiftsPerDay = 2;
34    int workingDays = 30 - 4;
35
36    int prodPerShift = KOL * hourPerShift;
37    int prodPerDay = prodPerShift * shiftsPerDay;
38    int prodPerMonth = prodPerDay * workingDays;
39
40    cout << "За 8-годинну зміну: " << prodPerShift << " шайб." << endl;
41    cout << "За робочий день: " << prodPerDay << " шайб." << endl;
42    cout << "За 30-денний місяць: " << prodPerMonth << " шайб." << endl;
43
44    return 0;

```



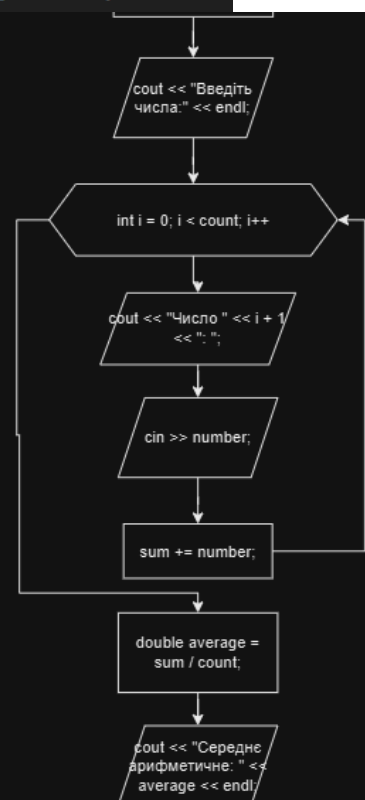
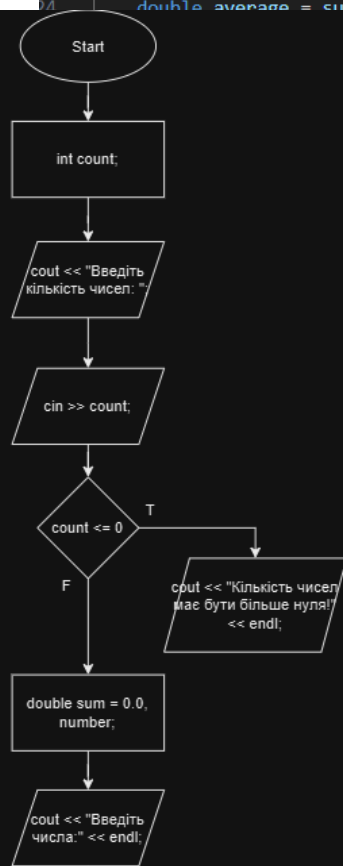
Завдання 4. Написати програму згідно свого варіанту.

Варіант 13. Скласти програму, яка обчислює середнє арифметичне послідовності дробових чисел, яка вводиться з клавіатури. Кількість чисел повинна задаватися під час роботи програми.

```

1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int count;
7      cout << "Введіть кількість чисел: ";
8      cin >> count;
9
10     if (count <= 0) {
11         cout << "Кількість чисел має бути більше нуля!" << endl;
12         return 1;
13     }
14
15     double sum = 0.0, number;
16
17     cout << "Введіть числа:" << endl;
18     for (int i = 0; i < count; i++) {
19         cout << "Число " << i + 1 << ": ";
20         cin >> number;
21         sum += number;
22     }
23
24     double average = sum / count;
25     cout << "Середнє арифметичне: " << average << endl;

```



Algotester task 1

Загадкове число

Обмеження: 2 сек., 256 МБ

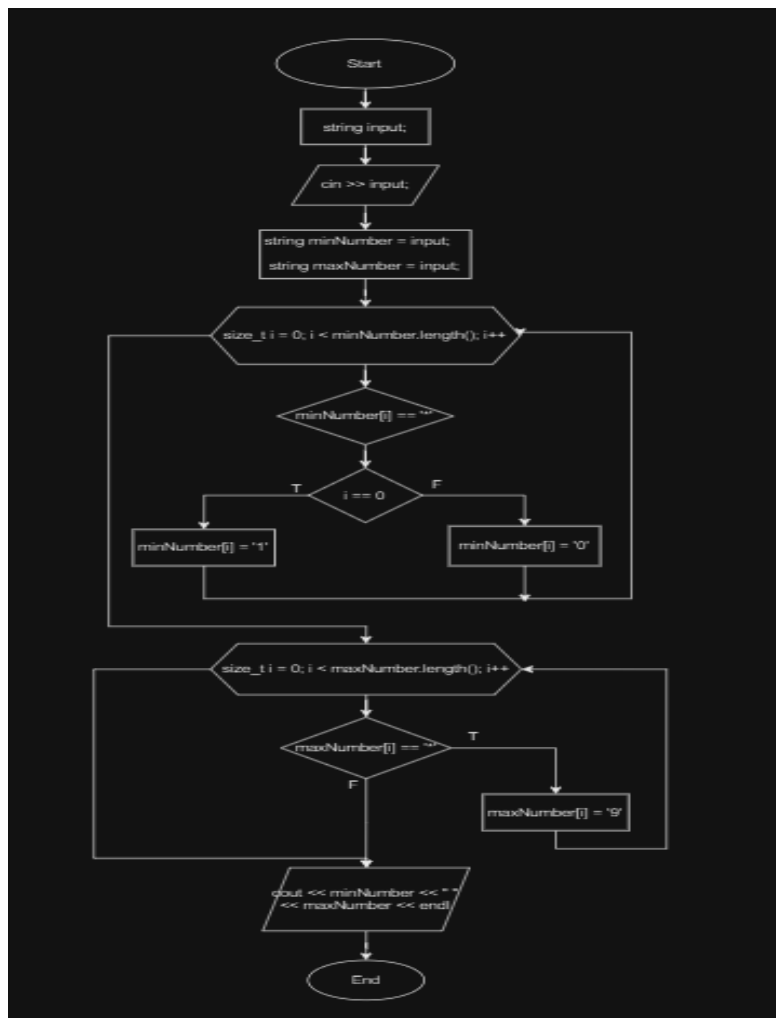
Марічка, як і годиться усім представницям прекрасної статі, полюбляє говорити загадками. Так і цього разу, Марічка планувала написати Зенику записку з її улюбленим числом, проте вона не втрималася, і деякі цифри в числі замінила на зірочки (символ `*`).

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

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

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  // Загадкове число
7  int main() {
8      string input;
9      cin >> input;
10
11     string minNumber = input;
12     string maxNumber = input;
13
14     for (size_t i = 0; i < minNumber.length(); i++) {
15         if (minNumber[i] == '*') {
16             if (i == 0) {
17                 minNumber[i] = '1';
18             } else {
19                 minNumber[i] = '0';
20             }
21         }
22     }
23
24     for (size_t i = 0; i < maxNumber.length(); i++) {
25         if (maxNumber[i] == '*') {
26             maxNumber[i] = '9';
27         }
28     }
29
30     cout << minNumber << " " << maxNumber << endl;
31
32     return 0;
33 }
34
```



Algotester task 2

Кредитні картки

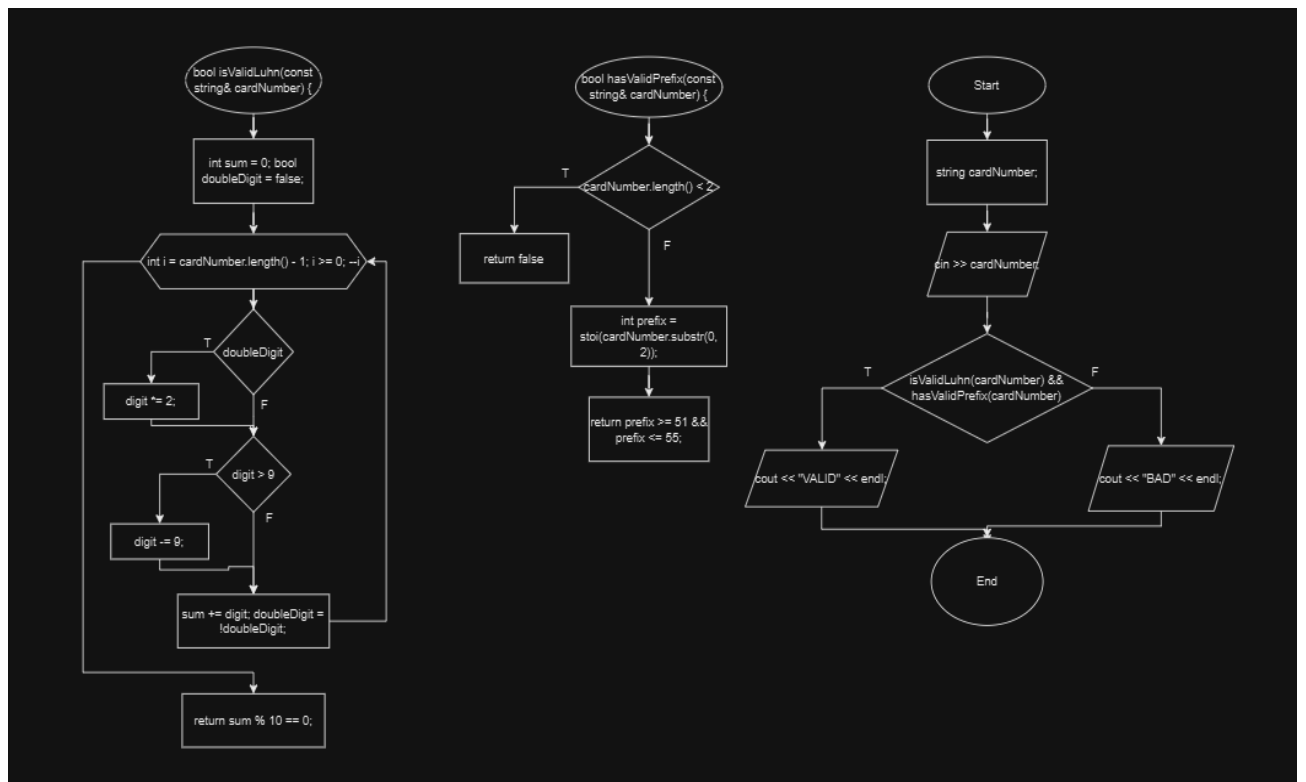
Обмеження: 2 сек., 256 MiB

Чи знали ви, що не всі номери кредитних карток є валідними? Для того, щоб уникнути хоча б частини помилок при введенні номерів кредитних карток, номери повинні задовільняти алгоритм Луна. Алгоритм Луна — це проста формула перевірки контрольної суми, котру використовують для валідації різноманітних ідентифікаційних номерів. Алгоритм працює наступним чином:

1. Дано певне додатнє число.
2. Помножимо кожену цифру на непарній позиції на 2 (тобто першу, третю і т.д.) і замінімо початкові цифри на ці результати. Якщо в результаті множення вийшло двоцифрове число, то випишемо обидві ці цифри.
3. Додамо разом усі цифри з числа, яке вийшло.
4. Якщо результат ділиться на 10 без остачі, то число з кроку 1 вважається валідним, в протилежному випадку це число не валідне.

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

```
1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  bool isValidLuhn(const string& cardNumber) {
7      int sum = 0;
8      bool doubleDigit = false;
9
10     for (int i = cardNumber.length() - 1; i >= 0; --i) {
11         int digit = cardNumber[i] - '0';
12
13         if (doubleDigit) {
14             digit *= 2;
15             if (digit > 9) {
16                 digit -= 9;
17             }
18         }
19
20         sum += digit;
21         doubleDigit = !doubleDigit;
22     }
23
24     return sum % 10 == 0;
25 }
26
27 bool hasValidPrefix(const string& cardNumber) {
28     if (cardNumber.length() < 2) {
29         return false;
30     }
31     int prefix = stoi(cardNumber.substr(0, 2));
32     return prefix >= 51 && prefix <= 55;
33 }
34
35 int main() {
36     string cardNumber;
37     cin >> cardNumber;
38
39     if (isValidLuhn(cardNumber) && hasValidPrefix(cardNumber)) {
40         cout << "VALID" << endl;
41     } else {
42         cout << "BAD" << endl;
43     }
44
45     return 0;
46 }
```



Algotester task 3

Літня школа

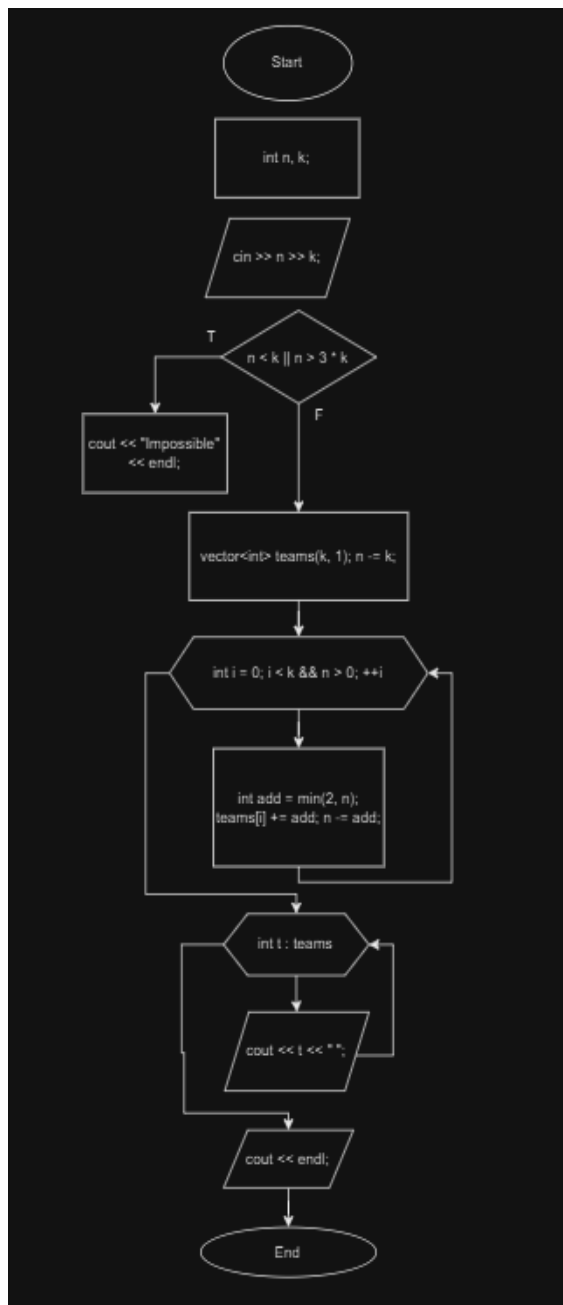
Обмеження: 2 сек., 256 MiB

Одного разу до Ужгорода на літню школу з алгоритмічного програмування приїхали n студентів, що сформували k команд. Відомо, що кожна команда складається з одного, двох або трьох студентів.

Вам необхідно визначити, скільки студентів було в кожній із команд.

C++ 23	Accepted	0.002	1.039
--------	----------	-------	-------

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main() {
6      int count;
7      cout << "Введіть кількість чисел: ";
8      cin >> count;
9
10     if (count <= 0) {
11         cout << "Кількість чисел має бути більше нуля!" << endl;
12         return 1;
13     }
14
15     double sum = 0.0, number;
16
17     cout << "Введіть числа:" << endl;
18     for (int i = 0; i < count; i++) {
19         cout << "Число " << i + 1 << ": ";
20         cin >> number;
21         sum += number;
22     }
23
24     double average = sum / count;
25     cout << "Середнє арифметичне: " << average << endl;
26
27     return 0;
28 }
```



Algotester task 4

Ліфт

Обмеження: 2 сек., 256 MiB

П'ятнадцятиповерхові - з цегли, дев'ятиповерхові - з бетону -

Однакові будинки вздовж і впоперек мого району.

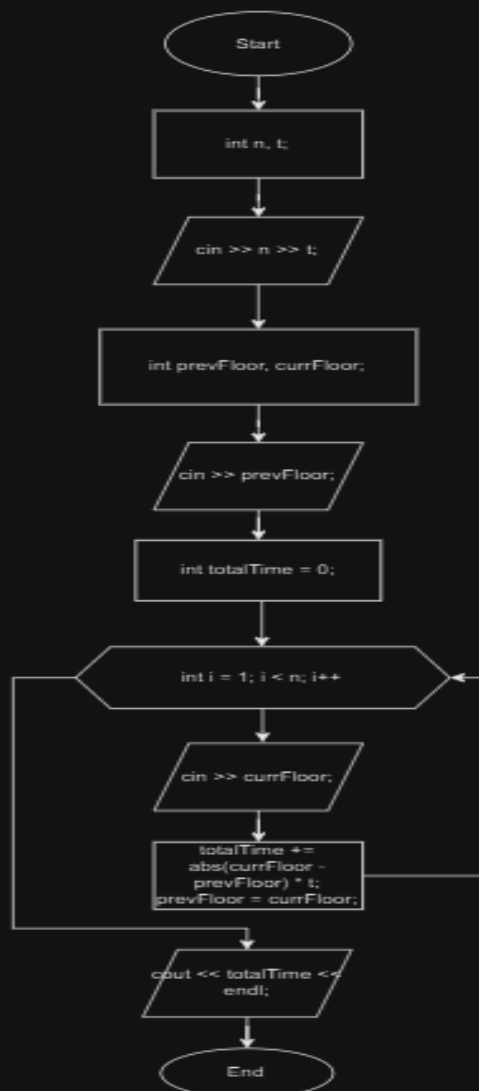
Так колись говорив один львівський митець з району на окраїні міста. Сьогодні його послідовник досліджує ліфти в цих будинках. Він підіймається на певний поверх пішки, а потім їздить туди-сюди між поверхами поки ліфт не зламається або йому не набридне. Вам потрібно знайти скільки сумарно часу тривала поїздка.

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

```

1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  int main() {
7      int n, t;
8      cin >> n >> t;
9
10     int prevFloor, currFloor;
11     cin >> prevFloor;
12
13     int totalTime = 0;
14     for (int i = 1; i < n; i++) {
15         cin >> currFloor;
16         totalTime += abs(currFloor - prevFloor) * t;
17         prevFloor = currFloor;
18     }
19
20     cout << totalTime << endl;
21     return 0;
22 }
23

```



Висновок:

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