

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  
Стик Назарій Олегович

Львів 2024

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

## Виконання роботи:

- Завдання №1: VNS Practice work 1 (Варіант-17)

Розробити лінійний алгоритм для розв'язання задачі.

**Варіант 17.**  $Y = b^3 \operatorname{tg}^2 x - \frac{a}{\sin^2(x/a)}$ ;  $Z = ae^{-\sqrt{3}} \cos(bx/a)$ , де  $a=3,2$ ;  $b=17,5$ ;  $x=-4,8$ .

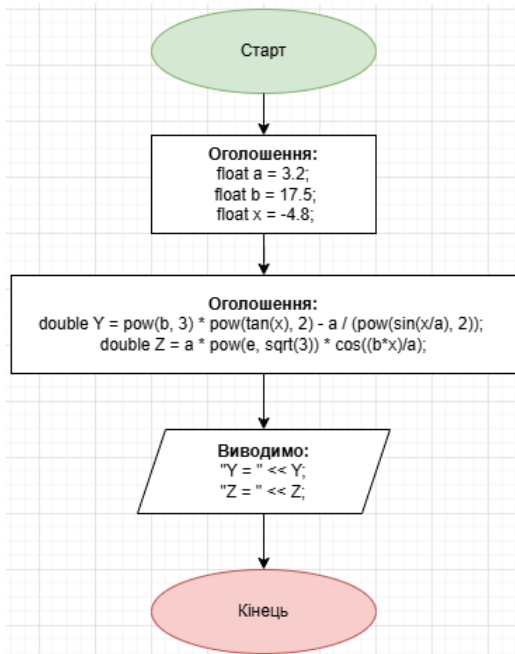
### Код:

```
1 // Завдання 1. Варіант 17
2 #include <iostream>
3 #include <cmath>
4
5 #define e 2.7182
6
7 using namespace std;
8
9 int main()
10 {
11     float a = 3.2;
12     float b = 17.5;
13     float x = -4.8;
14
15     double Y = pow(b, 3) * pow(tan(x), 2) - a / (pow(sin(x/a), 2));
16     double Z = a * pow(e, sqrt(3)) * cos((b*x)/a);
17
18     cout << "Y = " << Y << endl;
19     cout << "Z = " << Z << endl;
20
21     return 0;
22 }
```

### Результат виконання:

```
Y = 694651
Z = 7.92444
PS D:\VS Code\Projects>
```

## Блок-схема:



Очікувано часу: **30 хвилин.**

Витрачено часу: **15 хвилин.**

- **Завдання №2: VNS Practice work 2 (Варіант-21)**

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

**Варіант 21.** Увести п'ять наборів сторін трикутника  $a$ ,  $b$ ,  $c$  і визначити, для яких сторін висота  $h_a$  буде найбільшою. Підказка:

$$h_a = \frac{2}{a} \sqrt{p(p-a)(p-c)}, \text{ де } p = (a+b+c)/2.$$

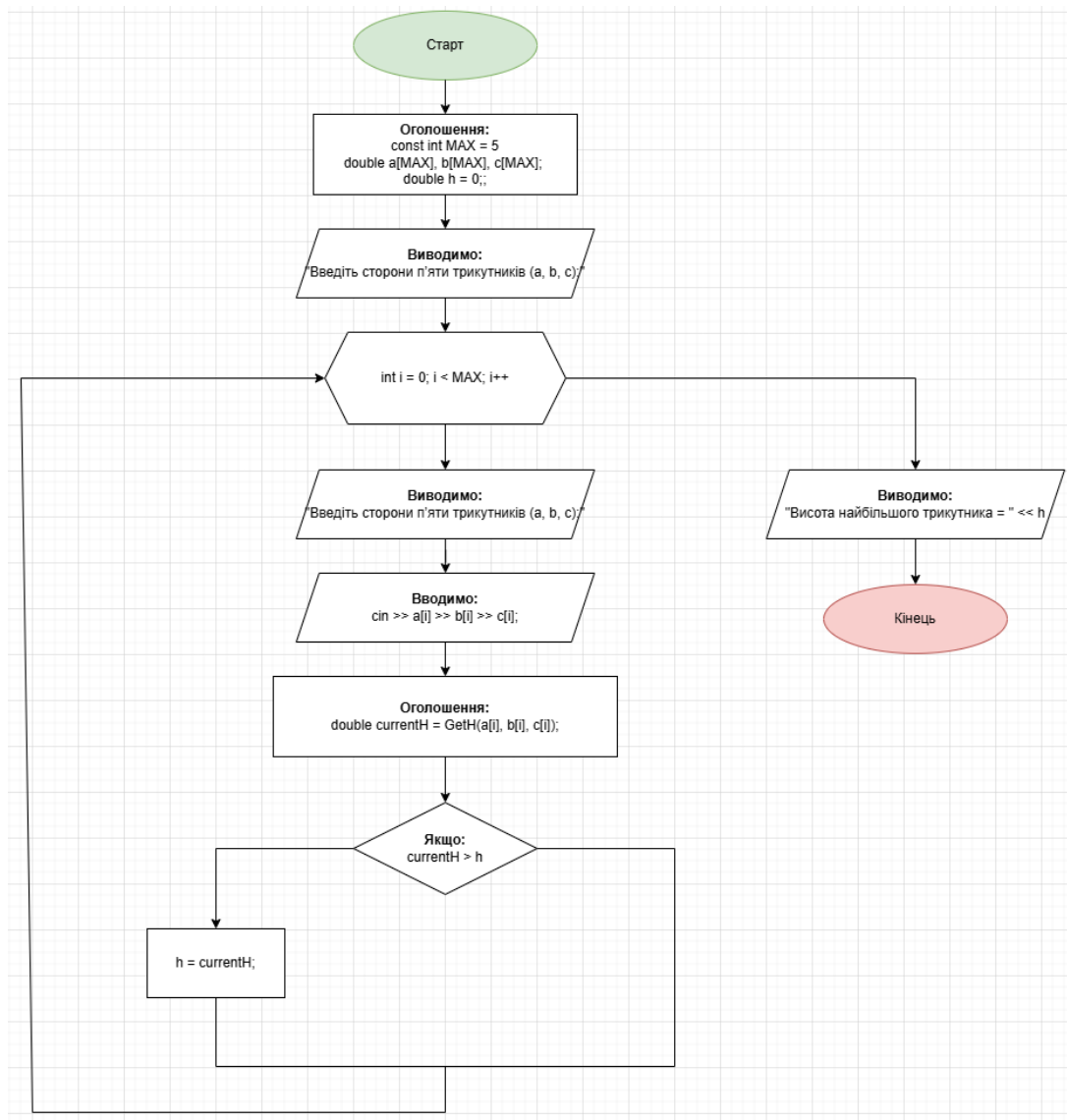
## Код:

```
1 // Варіант 21
2 #include <iostream>
3 #include <cmath>
4 #include <vector>
5
6 using namespace std;
7
8 double GetH(double a, double b, double c)
9 {
10     if (a <= 0)
11     {
12         return 0;
13     }
14
15     double p = (a + b + c) / 2;
16     return (2/a) * sqrt(p * (p - a) * (p - c));
17 }
18 int main()
19 {
20     const int MAX = 5;
21     double a[MAX], b[MAX], c[MAX];
22     double h = 0;
23
24     cout << "Введіть сторони п'яти трикутників (a, b, c):" << endl;
25     for (int i = 0; i < MAX; i++)
26     {
27         cout << "Трикутник " << i + 1 << ": ";
28         cin >> a[i] >> b[i] >> c[i];
29
30         double currentH = GetH(a[i], b[i], c[i]);
31         if (currentH > h)
32         {
33             h = currentH;
34         }
35     }
36
37     cout << "Висота найбільшого трикутника = " << h << endl;
38
39     return 0;
40 }
```

## Результат виконання:

```
Введіть сторони п'яти трикутників (a, b, c):
Трикутник 1: 2.5 3.4 5.7
Трикутник 2: 1.0 2.0 2.0
Трикутник 3: 8.4 9.7 5.6
Трикутник 4: 5.0 7.0 8.6
Трикутник 5: 9.1 8.2 5.8
Висота найбільшого трикутника = 3.85337
PS D:\VS Code\Projects>
```

## Блок-схема:



Очікувано часу: **30 хвилин.**

Витрачено часу: **45 хвилин.**

- **Завдання №3: VNS Practice work 3 (Варіант-2)**

Написати програму згідно свого варіанту.

**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).

## Код:

```
1 // Варіант - 2
2 #include <iostream>
3
4 using namespace std;
5
6 double Area(double length, double width, double height)
7 {
8     return 2*(length * width + length * height + width * height);
9 }
10
11 double ConvertToVerst(double km)
12 {
13     double m = km * 1000;
14
15     return m / 1066.8;
16 }
17
18 int main()
19 {
20     double length, width, height;
21
22     cout << "Обчислення площі поверхні паралелепіпеда" << endl;
23     cout << "Введіть початкові дані:" << endl;
24     cout << "Довжина (см) > ";
25     cin >> length;
26     cout << "Ширина (см) > ";
27     cin >> width;
28     cout << "Висота (см) > ";
29     cin >> height;
30     cout << "Площа поверхні: " << Area(length, width, height) << " кв.см." << endl;
31     cout << endl;
32
33
34
35     double km;
36     cout << "Перерахунок відстані з кілометрів у версти:" << endl;
37     cout << "Довжина (км) > ";
38     cin >> km;
39     cout << "Довжина у верстах: " << ConvertToVerst(km) << endl;
40
41
42     return 0;
43 }
```

## Результат виконання:

Обчислення площі поверхні паралелепіпеда

Введіть початкові дані:

Довжина (см) > 5.4

Ширина (см) > 8.7

Висота (см) > 4.5

Площа поверхні: 220.86 кв.см.

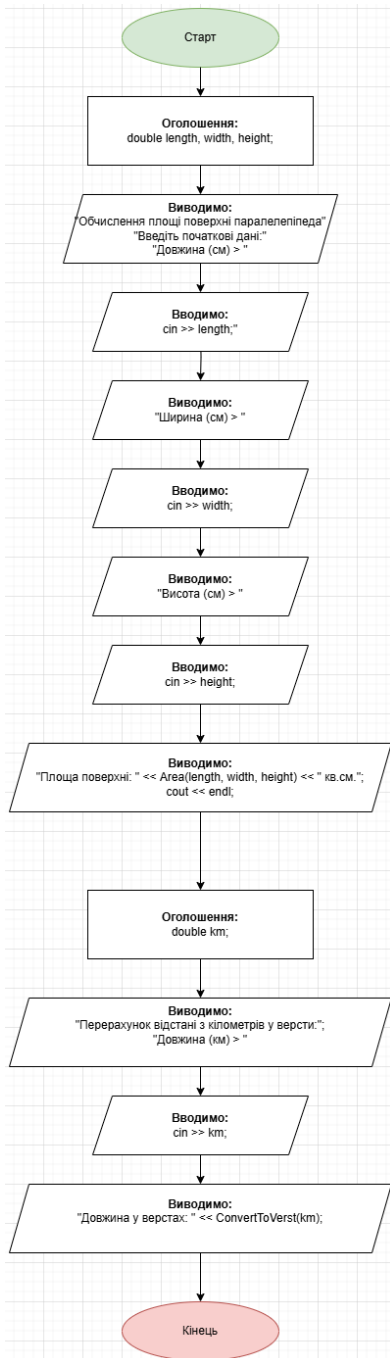
Перерахунок відстані з кілометрів у версти:

Довжина (км) > 5

Довжина у верстах: 4.68691

PS D:\VS Code\Projects> █

## Блок-схема:



Очікувано часу: **30 хвилин.**

Витрачено часу: **45 хвилин.**

- **Завдання №4: VNS Practice work 4 (Варіант-14)**

Написати програму згідно свого варіанту.

**Варіант 14.** Скласти програму, яка вводить з клавіатури послідовність з п'яти дробових чисел і після введення кожного числа виводить середнє арифметичне введеної частини послідовності.

**Код:**

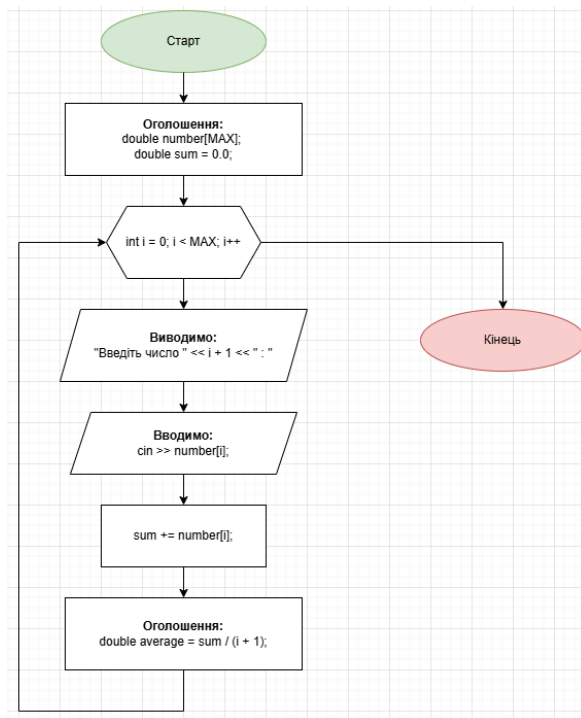
```
1 // Варіант 14
2 #include <iostream>
3 #define MAX 5
4
5 using namespace std;
6
7 int main()
8 {
9     double number[MAX];
10    double sum = 0.0;
11
12    for (int i = 0; i < MAX; i++)
13    {
14        cout << "Введіть число " << i + 1 << " : ";
15        cin >> number[i];
16
17        sum += number[i];
18        double average = sum / (i + 1);
19        cout << "Середнє значення чисел = " << average << endl;
20    }
21
22    return 0;
23 }
```

**Результат виконання:**

```
Введіть число 1 : 4.5
Середнє значення чисел = 4.5
Введіть число 2 : 2.8
Середнє значення чисел = 3.65
Введіть число 3 : 4.0
Середнє значення чисел = 3.76667
Введіть число 4 : 10.1
Середнє значення чисел = 5.35
Введіть число 5 : 15.4
Середнє значення чисел = 7.36
PS D:\VS Code\Projects>
```



## Блок-схема:



Очікувано часу: **30 хвилин.**

Витрачено часу: **30 хвилин.**

### • Завдання №5: Algotester task 1

#### Депутатські гроші

Обмеження: 2 сек., 256 MiB

Часто-густо громадяни намагаються з'ясувати, наскільки багатими є депутати. Дехто вірить, що матеріальні статки окремих депутатів є необмеженими.

Тож узяті собі депутата, у якого є необмежена кількість купюр усіх номіналів (1, 2, 5, 10, 20, 50, 100, 200 та 500 гривень). Він хоче придбати подарунок для своєї тещі, що коштує  $n$  гривень.

Незважаючи на те, що наш герой-олігарх є нескінченно багатим, він також є нескінченно скупим та педантичним. Саме тому він хоче оплатити покупку готівкою без решти. Яка мінімальна кількість купюр йому для цього знадобиться?

#### Вхідні дані

У єдиному рядку задано одне натуральне число  $n$  — вартість подарунку.

#### Вихідні дані

У єдиному рядку виведіть одне ціле число — мінімальну кількість купюр, що необхідна для покупки подарунка.

#### Обмеження

$1 \leq n \leq 10^9$ .

#### Приклади

Вхідні дані ( <i>stdin</i> )	Вихідні дані ( <i>stdout</i> )
74	4

#### Примітки

Подарунок, що коштує 74 гривні, можна оплатити п'ятдесятигривневою, двадцятигривневою та двома двогривневими купюрами.

Посилання: <https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40261>

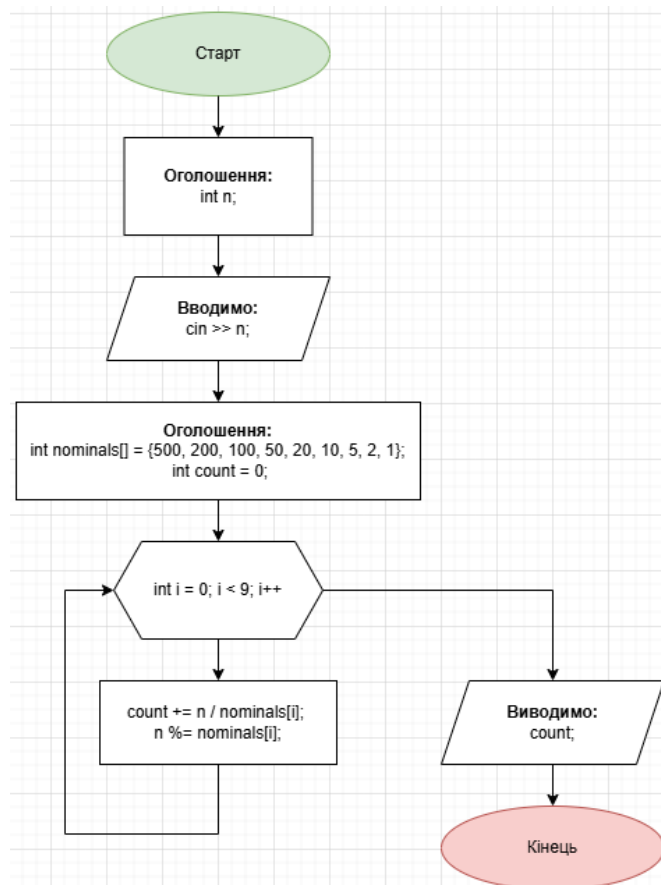
## Код:

```
1  // Депутатські гроші
2  #include <iostream>
3  using namespace std;
4
5  int main()
6  {
7      int n;
8      cin >> n;
9
10     int nominals[] = {500, 200, 100, 50, 20, 10, 5, 2, 1};
11     int count = 0;
12
13     for (int i = 0; i < 9; i++)
14     {
15         count += n / nominals[i];
16         n %= nominals[i];
17     }
18
19     cout << count << endl;
20     return 0;
21 }
```

## Результат виконання:

```
74
4
PS D:\VS Code\Projects> █
```

## Блок-схема:



Очікувано часу: **15 хвилин.**

Витрачено часу: **30 хвилин.**

- **Завдання №6: Algotester task 2**

### Марічка і печиво

Обмеження: 2 сек., 256 MiB

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 підглядела, 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.

### Обмеження

20% тестів:

$$1 \leq n \leq 1000, 0 \leq a_i \leq 10^4$$

60% тестів:

$$1 \leq n \leq 10^5, 0 \leq a_i \leq 10^4$$

20% тестів:

$$1 \leq n \leq 10^5, 0 \leq a_i \leq 10^9$$

### Приклади

Вхідні дані ( <i>stdin</i> )	Вихідні дані ( <i>stdout</i> )
4	128
4 7 47 74	

Посилання: <https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/2>

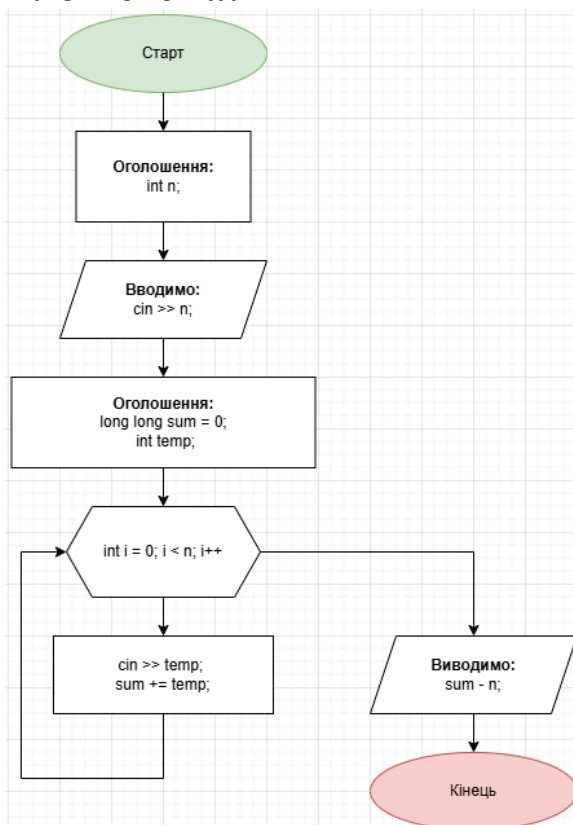
## Код:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int n;
7      cin >> n;
8
9      long long sum = 0;
10     int temp;
11
12     for (int i = 0; i < n; i++)
13     {
14         cin >> temp;
15         sum += temp;
16     }
17
18     cout << sum - n << endl;
19     return 0;
20 }
```

## Результат виконання:

```
4
4 7 47 74
128
PS D:\VS Code\Projects>
```

## Блок-схема:



Очікувано часу: **30 хвилин.**  
Витрачено часу: **30 хвилин.**

## • Завдання №7: Algotester task 3

Коля, Вася і Теніс

Обмеження: 2 сек., 256 MiB

Коли Коля та Вася прийшли робити ремонт на «Екстралогіку» — першим, що вони побачили в офісі, був стіл для настільного тенісу. Поки всі інші працювали, Коля та Вася вирішили пограти. Через декілька годин прийшов директор і накричав на заробітчан через те, що вони нічим не займаються. Тож Вася і Коля мушили йти працювати.

По дорозі вони сперечалися, хто ж виграв і з яким рахунком. Оскільки вони записували результати кожної подачі, то це можна порахувати. Але оскільки гра тривала дуже довго — порахувати це вручну дуже тяжко.

Всього відбулося  $n$  подач. Про кожну з них ми знаємо, хто переміг. За виграну подачу гравець отримує одне очко. Партія вважається виграною, коли один з гравців набере не менше одинадцяти очок з перевагою щонайменше у два очки. Наприклад, за рахунків 11:9, 4:11, 15:13 партія закінчується, а за рахунків 11:10 та 99:98 — ні. Як тільки Коля і Вася закінчили одну партію — вони починають іншу.

Знаючи, хто переміг кожної подачі — виведіть загальний рахунок по партіях в грі Коля-Вася. А якщо вони не дограли останню партію, то і її рахунок теж.

### Вхідні дані

У першому рядку задано ціле число  $n$  — загальна кількість подач.

У другому рядку задано  $n$  символів  $c_i$ .  $c_i = K$ , якщо  $i$ -ту подачу виграв Коля, та  $c_i = V$ , якщо  $i$ -ту подачу виграв Вася.

### Вихідні дані

У першому рядку виведіть загальний рахунок гри по партіях у форматі  $k : v$ , де  $k$  — кількість партій, у яких переміг Коля, а  $v$  — кількість партій, у яких переміг Вася.

Якщо вони не дограли останню партію, то в другому рядку в такому ж форматі виведіть рахунок останньої партії.

### Обмеження

30% тестів:  $1 \leq n \leq 10^4$

70% тестів:  $1 \leq n \leq 10^5$

### Приклади

Вхідні дані ( <i>stdin</i> )	Вихідні дані ( <i>stdout</i> )
30 VVVKVKKVVVVVKVKKKKVVKKVKKKKVVVV	1:0 2:4

### Примітки

Першу партію виграв Коля з рахунком 13:11, а другу вони не встигли дограти. В ній рахунок був 2:4.

### Посилання:

<https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/13>

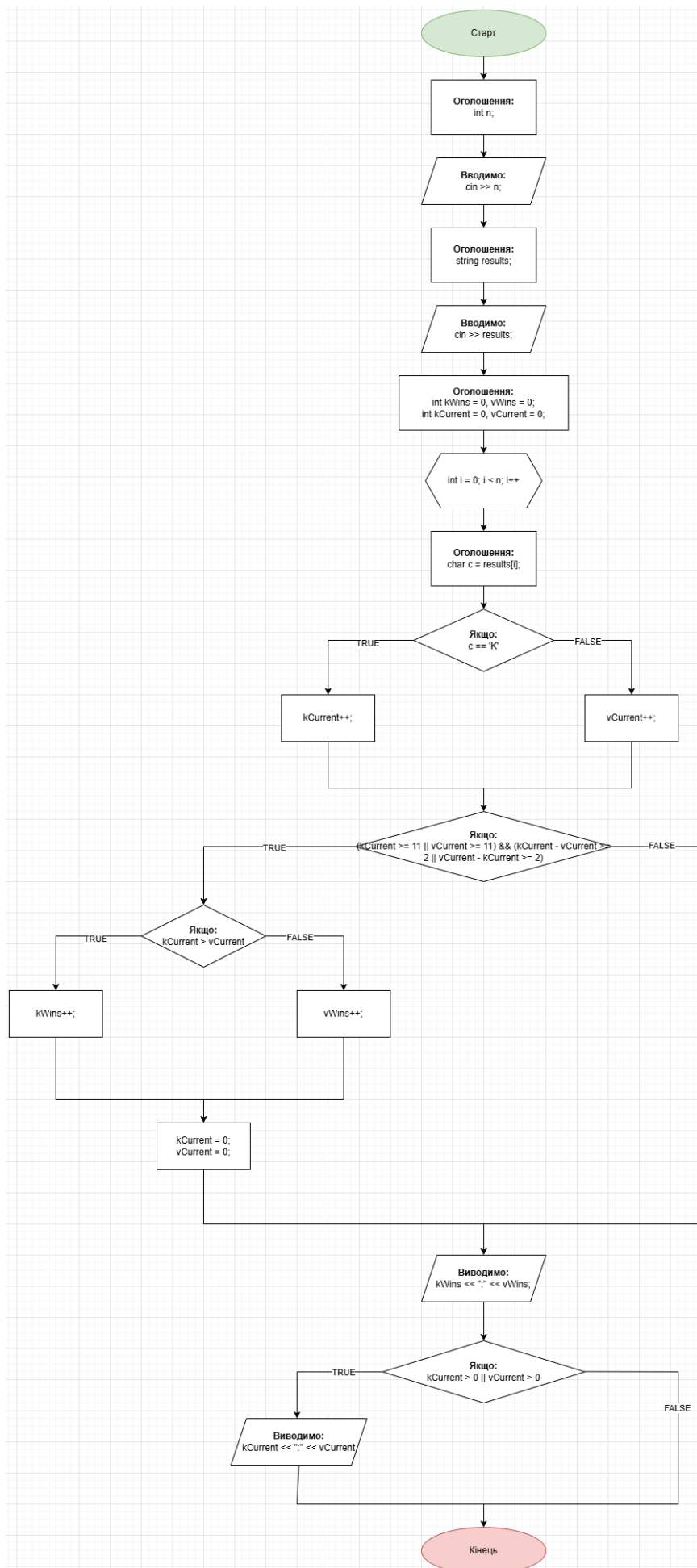
## Код:

```
1 // Коля, Вася і Теміс
2 #include <iostream>
3 using namespace std;
4
5 int main()
6 {
7     int n;
8     cin >> n;
9
10    string results;
11    cin >> results;
12
13    int kWins = 0, vWins = 0;
14    int kCurrent = 0, vCurrent = 0;
15
16    for (int i = 0; i < n; i++)
17    {
18        char c = results[i];
19        if (c == 'K')
20        {
21            kCurrent++;
22        }
23        else
24        {
25            vCurrent++;
26        }
27
28        if ((kCurrent >= 11 || vCurrent >= 11) && (kCurrent - vCurrent >= 2 || vCurrent - kCurrent >= 2))
29        {
30            if (kCurrent > vCurrent)
31            {
32                kWins++;
33            }
34            else
35            {
36                vWins++;
37            }
38
39            kCurrent = 0;
40            vCurrent = 0;
41        }
42    }
43
44    cout << kWins << ":" << vWins << endl;
45
46    if (kCurrent > 0 || vCurrent > 0)
47    {
48        cout << kCurrent << ":" << vCurrent << endl;
49    }
50
51    return 0;
52 }
```

## Результат виконання:

```
30
VKVKKVVVKVKKKKKVKKVKKKKVVV
1:0
2:4
PS D:\VS Code\Projects> █
```

## Блок-схема:



Очікувано часу: **30 хвилин.**  
Витрачено часу: **1,5 години.**

- **Завдання №8: Algotester task 4**

### Факторіал

Обмеження: 1 сек., 256 MiB

Підрахуйте кількість нулів у кінці запису факторіалу числа  $n$ .

### Вхідні дані

У єдиному рядку задано одне ціле число  $n$ .

### Вихідні дані

У єдиному рядку виведіть одне ціле число — кількість нулів у кінці числа.

### Обмеження

$1 \leq n \leq 2 \times 10^9$ .

### Приклади

Вхідні дані ( <i>stdin</i> )	Вихідні дані ( <i>stdout</i> )
8	1

Посилання: <https://algotester.com/uk/ArchiveProblem/DisplayWithEditor/40627>

### Код:

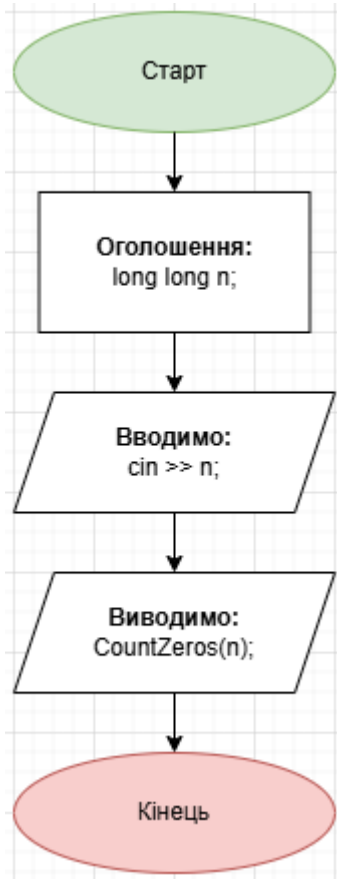
```
1 // Факторіал
2 #include <iostream>
3 using namespace std;
4
5 int CountZeros(long long n)
6 {
7     int count = 0;
8
9     while (n >= 5)
10    {
11        n /= 5;
12        count += n;
13    }
14
15    return count;
16 }
17
18 int main()
19 {
20     long long n;
21     cin >> n;
22     cout << CountZeros(n) << endl;
23
24     return 0;
25 }
```



## Результат виконання:

```
8  
1  
PS D:\VS Code\Projects> |
```

## Блок-схема:



Очікувано часу: **45 хвилин.**

Витрачено часу: **30 хвилин.**

## Pull-Request:

**Висновок:** Під час виконання цієї роботи я здобув практичні навички у створенні та дослідженні алгоритмів для розв'язання задач. Це допомогло мені краще зрозуміти, як застосовувати теорію на практиці, і покращило мої вміння.