

Міністерство освіти і науки України  
Національний університет «Львівська політехніка»  
Кафедра систем штучного інтелекту



## Звіт

**про виконання лабораторних та практичних робіт блоку № 2**

На тему: «Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.»

**з дисципліни:** «Основи програмування»

до:

ВНС Лабораторної Роботи № 1  
Алготестер Лабораторної Роботи № 1  
Практичних Робіт до блоку № 2

**Виконав:**

Студент групи ШІ-13  
Бойко Роман Андрійович

**Тема роботи:** Лінійні алгоритми. Розгалужені алгоритми. Умовні та логічні оператори. Змінні. Константи. Типи даних. Розмір типів даних. Ввід вивід. Базові операції та вбудовані функції. Коментарі.

**Мета роботи:** Навчитися використовувати логічні оператори для різних видів завдань, розібратися з різними типами даними та вбудованими функціями.

### Теоретичні відомості:

- умовні оператори if та switch
- ввід та вивід даних на мовах C/C++
- типи даних

Джерела

- <https://itproger.com/ua/course/cpp/4>
- <https://www.programiz.com/c-programming/c-input-output>
- <https://www.scholarhat.com/tutorial/cpp/data-types-in-cpp>
- Конспект лекцій з програмування О. Ю. Пшеничний

## Виконання роботи

### Завдання 1: VNS Lab 1

#### Особистий варіант - 25

1. Обчислити значення виразу при різних дійсних типах даних (float й double).

Обчислення варто виконувати з використанням проміжних змінних.

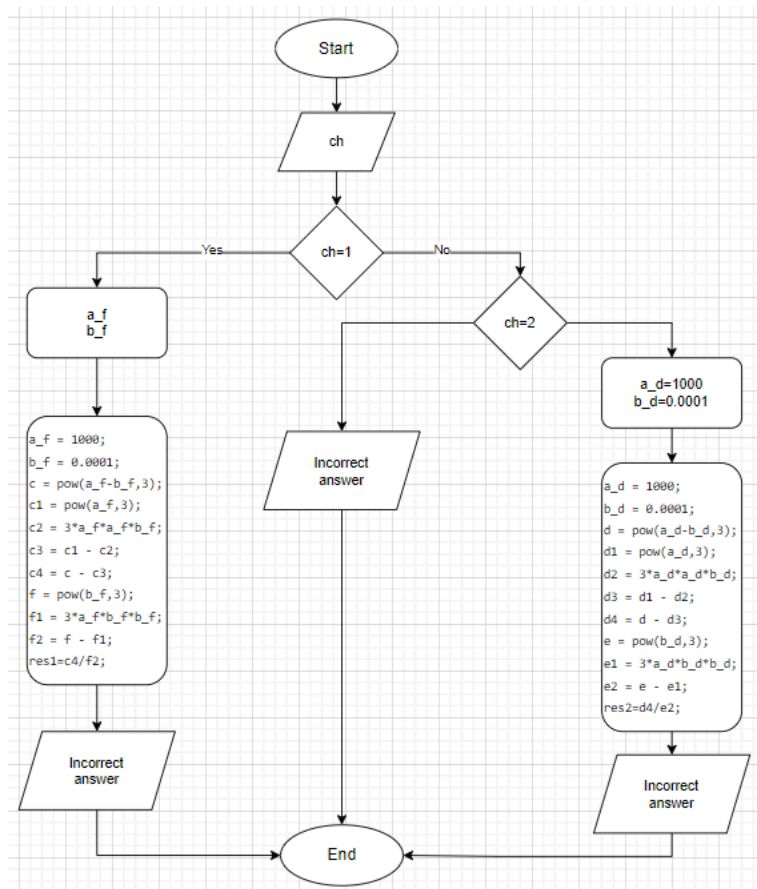
Порівняти й пояснити отримані результати.

$$\frac{(a-b)^3 - (a^3 - 3a^2b)}{b^3 - 3ab^2} \text{ при } a=1000, b=0.0001$$

2. Обчислити значення виразів. Пояснити отримані результати.

- 1) - -m-++n
- 2) m\*n<n++
- 3) n-- > m++

## Блок-схема до задачі:



## Розв'язок задачі

```

1  #include <iostream>
2  #include <cmath>
3  #include <iomanip>
4
5  using namespace std;
6
7  int main(){
8  int ch;
9  cout<<"Enter float - 1 or double - 2:";
10 cin>>ch;
11 if(ch==1){
12     float a_f = 1000;
13     float b_f = 0.0001;
14     float c = pow(a_f-b_f,3);
15     float c1 = pow(a_f,3);
16     float c2 = 3*a_f*a_f*b_f;
17     float c3 = c1 - c2;
18     float c4 = c - c3;
19     float f = pow(b_f,3);
20     float f1 = 3*a_f*b_f*b_f;
21     float f2 = f - f1;

```

```

22     float res1=c4/f2;
23     cout<<"Result in float: "<<setprecision(20)<<res1<<endl;
24 }else if(ch==2){
25     double a_d = 1000;
26     double b_d = 0.1;
27     double d = pow(a_d-b_d,3);
28     double d1 = pow(a_d,3);
29     double d2 = 3*a_d*a_d*b_d;
30     double d3 = d1 - d2;
31     double d4 = d - d3;
32     double e = pow(b_d,3);
33     double e1 = 3*a_d*b_d*b_d;
34     double e2 = e - e1;
35     double res2=d4/e2;
36     cout<<"Result in double: "<<setprecision(20)<<res2<<endl;
37 }
38 else{
39     cout<<"Incorrect answer"<<endl;
40 }
41
42 }

```

**Результат виконання програми:**

```

Enter float - 1 or double - 2:1
Result in float: 2133333.75

```

```

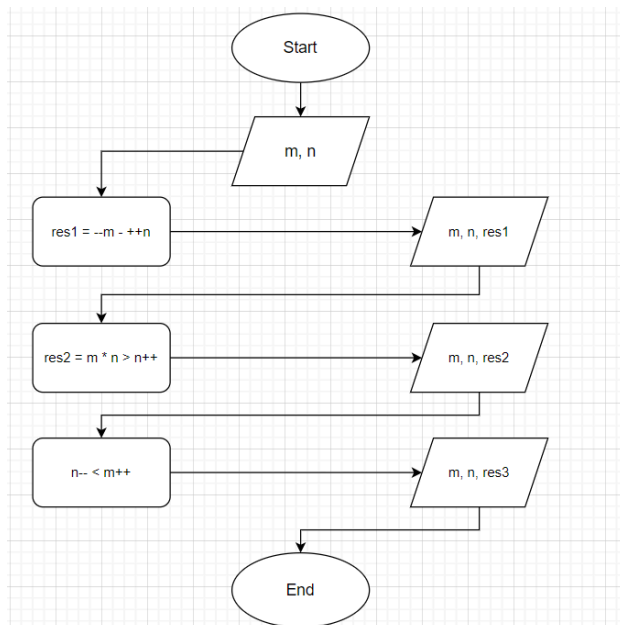
Enter float - 1 or double - 2:2
Result in double: -1.0013580656051646045

```

**Час виконання завдання ~ 20 хв**

**2)**

**Блок-схема до задачі:**



**Розв'язок задачі:**

```

#include <iostream>

using namespace std;

int main(){
    int m, n;
    cout<<"Enter m: ";
    cin>>m;
    cout<<"Enter n: ";
    cin>>n;
    int res1 = --m - ++n;
    cout<<"m: "<<m<<endl;
    cout<<"n: "<<n<<endl;
    cout<<"Result 1: "<<res1<<endl;
    int res2 = m * n > n++;
    cout<<"m: "<<m<<endl;
    cout<<"n: "<<n<<endl;
    cout<<"Result 2: "<<res2<<endl;
    int res3 = n-- < m++;
    cout<<"m: "<<m<<endl;
    cout<<"n: "<<n<<endl;
    cout<<"Result 3: "<<res3<<endl;
}

```

**Результат виконання програми:**

```

Enter m: 5
Enter n: 6
m: 4
n: 7
Result 1: -3
m: 4
n: 8
Result 2: 1
m: 5
n: 7
Result 3: 0

```

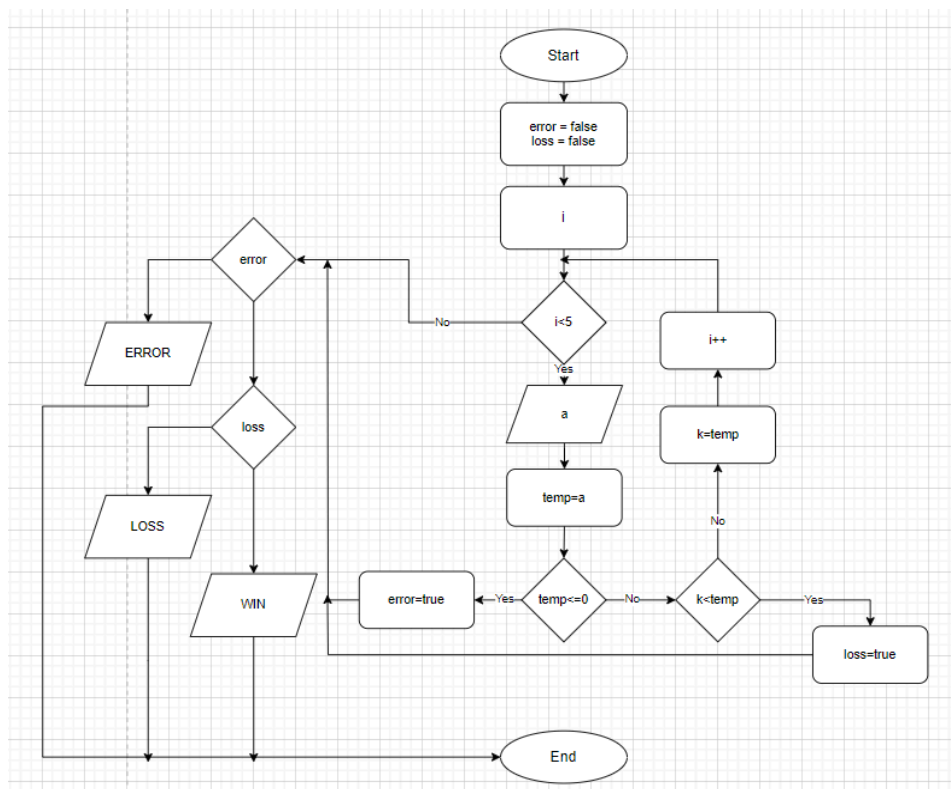
**Час виконання завдання ~ 10 хв**

## **Завдання 2: Algotester Lab 1**

### **Особистий варіант - 3**

Персонажу по одному дають сторони 5 кубів  $a_1..5$ , з яких він будує піраміду.

Коли він отримує куб з ребром  $a_i$  - він його ставить на існуючий, перший ставить на підлогу (вона безмежна).



### Розв'язок задачі:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main(){
6      long long int a, temp, k;
7      bool error = false;
8      bool loss = false;
9      for(int i = 0; i<5;i++){
10         cin>>a;
11         temp=a;
12         if(temp<=0){
13             error = true;
14             break;
15         }else if(k<temp){
16             loss = true;
17             break;
18         }
19         k=temp;
20     }
21     if(error){
22         cout<<"ERROR"<<endl;
23     }
24     else if(loss){
25         cout<<"LOSS"<<endl;
26     }else{
27         cout<<"WIN"<<endl;
28     }
29 }
```

### Результат виконання програми:

```
5
4
3
3
2
WIN
```

**Час виконання завдання ~ 20 хв**

### Завдання 3: Practice task

Ви створюєте простий порадник щодо погоди. Користувач вводить поточні погодні умови, а програма видає рекомендації щодо активності на основі погоди.

Можливі варіанти погоди:

- sunny;
- rainy;
- cloudy;
- snowy;
- windy;

## Мета Задачі

Навчитися користуватися операторами галуження для структурування логіки програм. Якщо користувач вводить будь-яку іншу умову, запропонуйте йому ввести дійсну умову.

## Використовуйте таку логіку

*if else* - щоб вирішити, чи повинен користувач взяти куртку чи ні.

*if, else if* - щоб надати рекомендацію щодо активності (прогулянка, футбол, настільні ігри, etc).

*switch case* - для визначення типу рекомендованого взуття.

## Деталі логіки

Рішення чи брати куртку (використовуючи *if else*)

- Якщо йде сніг або дощ, користувач повинен одягнути куртку.
- В іншому випадку куртка не потрібна.

Рекомендація щодо активності (використання *if, else if*)

- Якщо сонячно, порекомендуйте «Чудовий день для пікніка!».
- Інакше, якщо буде дощ, рекомендуємо «Ідеальна погода, щоб читати книгу всередині!».
- Інакше, якщо хмарно, рекомендуємо «Може, відвідати музей?».
- Інакше, якщо сніг, порекомендуйте «Як щодо того, щоб зліпити сніговика?».
- Інакше, якщо буде вітер, порекомендуйте «Запустіть повітряного змія, якщо він у вас є!».

Рекомендації щодо взуття (з використанням футляра для вимикача)

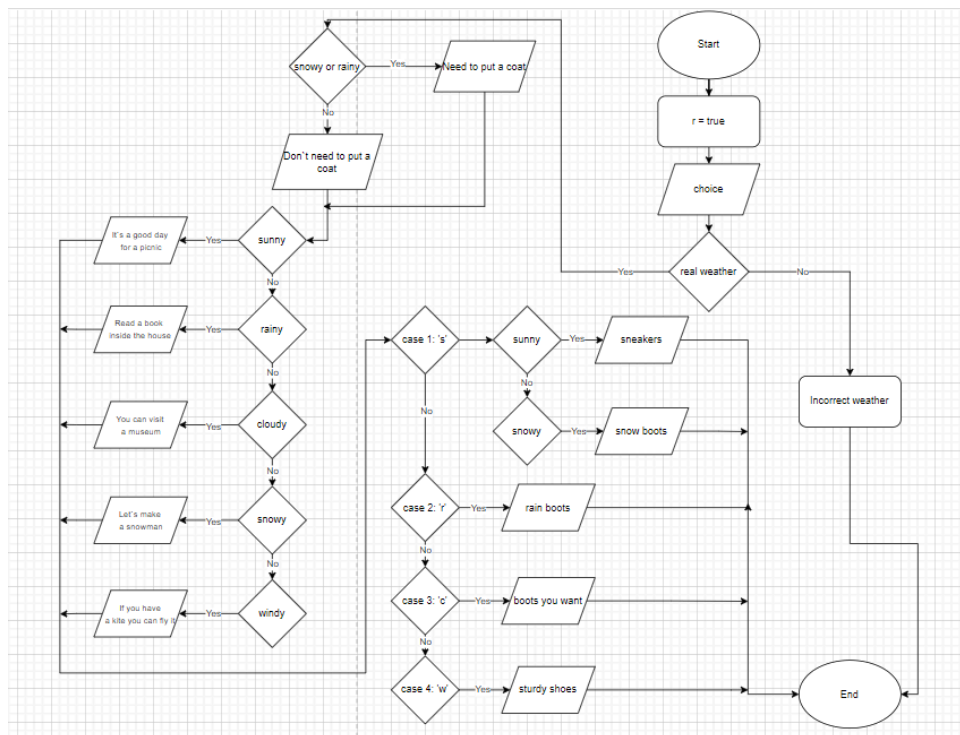
- sunny -> "Взуй улюблені кросівки!"
- rainy -> "Дощові чоботи - гарна ідея!"
- cloudy -> "Сьогодні підходить будь-яке взуття."
- snowy -> "Снігові черевики зігріють ваші ноги!"
- windy -> "Одягніть щось міцне!"

## Вимоги:

1. Використати всі згадані в передумові задачі оператори галуження - *if else, if, else if, switch case*;
2. За потреби комбінувати оператори;



## Блок-схема до задачі:



## Розв'язок задачі:

```

1  #include <iostream>
2  #include <string>
3  using namespace std;
4
5  int main(){
6      string choice;
7      bool r = true;
8      cout<<"Enter weather condition(sunny, rainy, cloudy, snowy, windy): ";
9      cin>>choice;
10     if(choice == "sunny" || choice == "rainy" || choice == "cloudy" || choice == "snowy" || choice == "windy"){
11
12         if (choice=="snowy" || choice=="rainy"){
13             cout<<"You should put on your coat!"<<endl;
14         }else{
15             cout<<"You don't have to put a coat!"<<endl;
16         }
17
18         if (choice == "sunny"){
19             cout<<"It's a good day for a picnic!"<<endl;
20         }else if(choice == "rainy"){
21             cout<<"It's recommended to read a book inside the house!"<<endl;
22         }else if(choice == "cloudy"){
23             cout<<"You can visit a museum!"<<endl;
24         }else if(choice == "snowy"){
25             cout<<"Let's make a snowman!"<<endl;
26         }else if(choice == "windy"){
27             cout<<"If you have a kite you can fly it!"<<endl;
28         }
29
30         switch (choice[0])
  
```

```

31     {
32         case 's':
33             if (choice == "sunny"){
34                 cout<<"Put your favourite sneakers!"<<endl;
35             }
36             else if(choice == "snowy"){
37                 cout<<"Snow boots will keep your feet warm!"<<endl;
38             }
39             break;
40         case 'r':
41             cout<<"Rain boots will be the best desicion!"<<endl;
42             break;
43         case 'c':
44             cout<<"You can put on boots you want!"<<endl;
45             break;
46         case 'w':
47             cout<<"You are recommended to put sturdy shoes!"<<endl;
48             break;
49         default:
50             break;
51     }
52
53 }else{
54     cout<<"You have enetered incorrect weather condition!"<<endl;
55 }
56 }

```

### Результат виконання програми:

```

Enter weather condition(sunny, rainy, cloudy, snowy, windy): sunny
You don't have to put a coat!
It's a good day for a picnic!
Put your favourite sneakers!

```

### Час виконання завдання ~ 25 хв

### Завдання 4: Self Algotester Task 1

У вашого персонажа є  $N$  хітпойнтів та  $M$  мани.

Персонаж 3 рази використає закляття, кожне з яких може використати хітпойнти та ману одночасно.

Якщо якесь закляття забирає і хітпойнти і ману - ваш персонаж програє, отже для виграшу треба використовувати при одному заклинанні АБО хітпойнти, АБО ману.

Якщо в кінці персонаж буде мати додатню кількість хітпойнтів та мани ( $N, M > 0$ ) - він виграє, в іншому випадку програє.

Ваше завдання у випадку виграшу персонажа вивести YES, вивести NO у іншому випадку.

Вхідні дані

2 цілих числа

$N$  та  $M$  - хітпойнти та мана персонажа

3 рядки по 2 цілих числа,

$h_i$  та  $m_i$  - кількість хітпойнтів та мани, які ваш персонаж потратить за хід на  $i$  заклинання

Вихідні дані

YES - якщо ваш персонаж виграє

NO - у всіх інших випадках

Обмеження

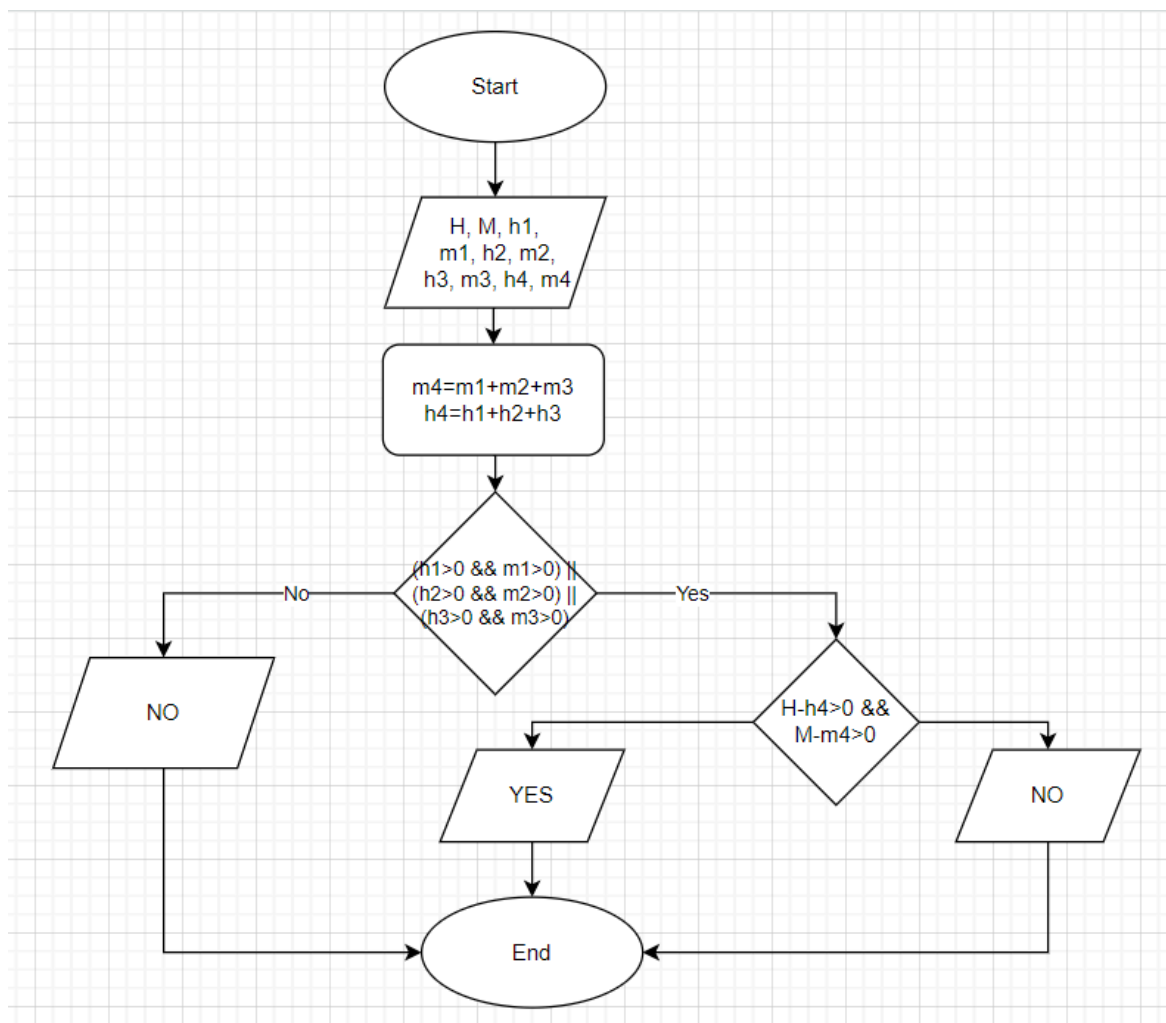
$1 \leq H \leq 10^{12}$

$1 \leq M \leq 10^{12}$

$0 \leq h_i \leq 10^{12}$

$0 \leq m_i \leq 10^{12}$

**Блок-схема до задачі:**



### Розв'язок задачі:

```
#include <iostream>

using namespace std;

int main(){
    long long int H, M, h1, m1, h2, m2, h3, m3, m4, h4;
    cin>>H>>M;
    cin>>h1>>m1;
    cin>>h2>>m2;
    cin>>h3>>m3;
    m4=m1+m2+m3;
    h4=h1+h2+h3;
    if((h1>0 && m1>0)|| (h2>0 && m2>0)|| (h3>0 && m3>0)){
        cout<<"NO"<<endl;
    }else if(H-h4>0 && M-m4>0){
        cout<<"YES"<<endl;
    }else{
        cout<<"NO"<<endl;
    }
}
```

### Результат виконання програми:

```
100000 50000
100 0
0 2000
1000 0
YES
```

**Час виконання завдання ~ 15 хв**

### Завдання 5: Self Algotester Task 2

#### Марічка і печиво

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:

$1 \leq n \leq 10^5$ ,

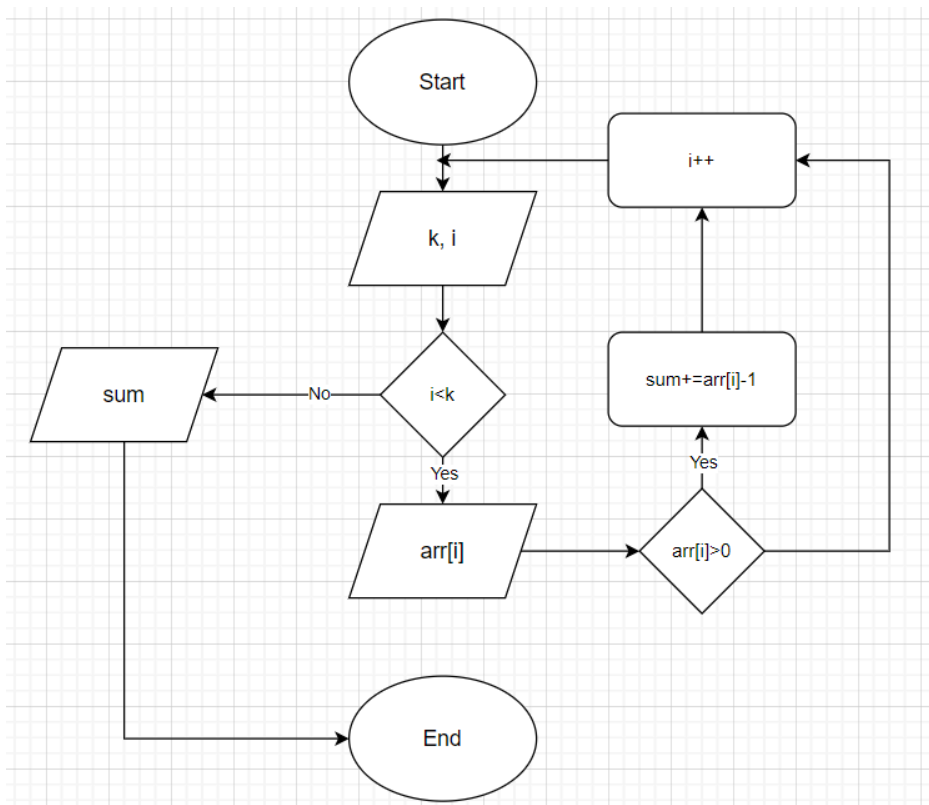
$0 \leq a_i \leq 10^4$

20% тестів:

$1 \leq n \leq 10^5$ ,

$0 \leq a_i \leq 10^9$

### Блок-схема до задачі:



## Розв'язок задачі:

```
#include <iostream>

using namespace std;

int main(){
    long long k, sum;
    cin>>k;
    int arr[k];
    for(int i = 0; i < k; i++){
        cin>>arr[i];
        if(arr[i]>0){
            sum+=arr[i]-1;
        }
    }
    cout<<sum<<endl;
}
```

## Результат виконання програми:

```
3
45
34
23
170
```

Час виконання завдання ~ 15 хв

Також налаштував файл launch.json

```
1 {
2     // Use IntelliSense to learn about possible attributes.
3     // Hover to view descriptions of existing attributes.
4     // For more information, visit: https://go.microsoft.com/fwlink/?linkid=830387
5     "version": "0.2.0",
6     "configurations": [
7         {
8             "name": "(gdb) Bash on Windows Launch",
9             "type": "cppdbg",
10            "request": "launch",
11            "program": "enter program name, for example ${workspaceFolder}/a.exe",
12            "args": [],
13            "stopAtEntry": false,
14            "cwd": "${fileDirname}",
15            "environment": [],
16            "externalConsole": false,
17            "pipeTransport": {
18                "debuggerPath": "/usr/bin/gdb",
19                "pipeProgram": "${env:windir}\\system32\\bash.exe",
20                "pipeArgs": ["-c"],
21                "pipeCwd": ""
22            },
23            "setupCommands": [
24                {
25                    "description": "Enable pretty-printing for gdb",
26                    "text": "-enable-pretty-printing",
27                    "ignoreFailures": true
28                },
29                {
30                    "description": "Set Disassembly Flavor to Intel",
31                    "text": "-gdb-set disassembly-flavor intel",
32                    "ignoreFailures": true
33                }
34            ]
35        }
36    ]
37 }
```

**Висновок:**

В цьому епіку я навчився користуватися оператором if-else та switch.

Навчився малювати перші блок-схеми до задач, розглянув цикли та зробив 2 додаткові задачі.