

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:

Практичних Робіт до блоку № 2

Виконав:

Студент групи ІІІ-13
Недосіка Назарій Вадимович

Львів 2024

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:

1. Змінні, Константи, Типи даних та їх Розміри:

- Визначення та оголошення змінних і констант.
- Основні типи даних у C++ (int, char, float, double, тощо).
- Розміри типів даних і їх представлення у двійковій системі.
- Порівняння різних типів даних та вибір відповідного типу для задач.

2. Бібліотеки в C++

- Поняття бібліотеки;
- Використання бібліотек;
- Бібліотеки: stdio, cmath, iostream, algorithm, vector, string.

3. Ввід та Вивід даних:

- Використання cin та cout.
- Форматування виводу даних.
- Робота з різними типами даних у вводі/виводі.
- Практичні вправи на ввід та вивід даних.

4. Базові Операції та Вбудовані Функції:

- Арифметичні операції та їх використання.
- Використання математичних функцій.
- Вправи на роботу з базовими операціями та функціями.

5. Лінійні алгоритми:

- Визначення та особливості лінійних алгоритмів.
- Структура та властивості лінійних алгоритмів.
- Написання лінійних алгоритмів на C++.
- Вправи на створення лінійних алгоритмів.

6. Розгалужені алгоритми та Умовні Оператори:

- Умовні оператори: if, else, else if, switch-case.
- Тернарний оператор ?.
- Реалізація розгалужених алгоритмів на C++.
- Практичні задачі на розгалужені алгоритми.

7. Логічні Оператори:

- Визначення та види логічних операторів (AND, OR, NOT).
- Застосування логічних операторів у розгалужених алгоритмах.
- Практичні приклади використання логічних операторів.
- Вправи на розуміння та використання логічних операцій.

Виконання роботи

Завдання 1: VNS Lab 1

Варіант – 12

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

Обчислення варто виконувати з використанням проміжних змінних.

Порівняти й пояснити отримані результати.

$$\frac{(a + b)^2 - (a^2 + 2ab)}{b^2}$$

При a=1000, b=0.0001

2. Обчислити значення виразів. Пояснити отримані результати.

1) - -m-++n

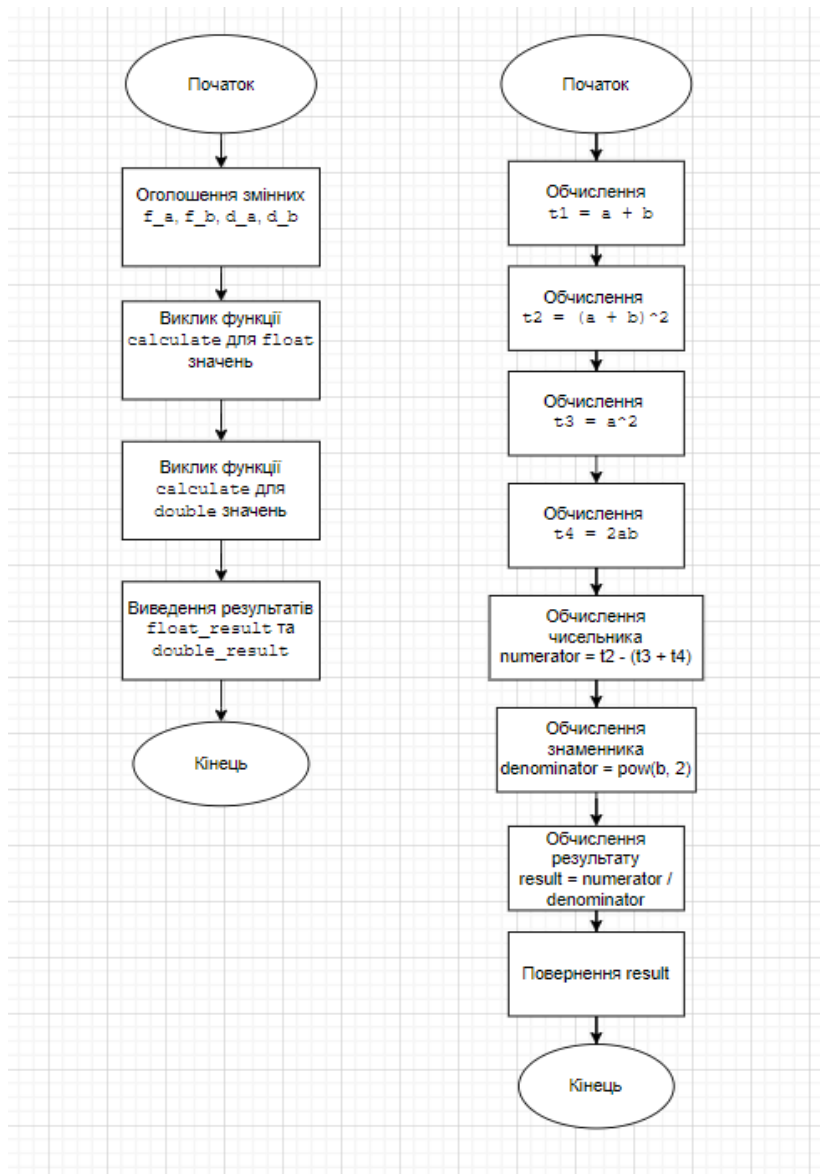
2) m*n<n++

3) n-- > m++

1.Розв'язок задачі:

```
1  #include <iostream>
2  #include <cmath>
3
4  using namespace std;
5
6  template <typename T>
7  T calculate(T a, T b)
8  {
9      T t1 = a + b;           // (a + b)
10     T t2 = pow(t1, 2);       // (a + b)^2
11     T t3 = pow(a, 2);        // a^2
12     T t4 = 2 * a * b;       // 2ab
13     T numerator = t2 - (t3 + t4); // (a+b)^2 - (a^2 + 2ab)
14     T denominator = pow(b, 2); // b^2
15     T result = numerator / denominator; // Остаточний результат
16     return result;
17 }
18
19 int main()
20 {
21     float f_a = 1000.0f, f_b = 0.0001f;           //6-9 цифр після коми
22     float float_result = calculate(f_a, f_b);
23
24     double d_a = 1000.0, d_b = 0.0001;           //15-18 цифр після коми
25     double double_result = calculate(d_a, d_b);
26
27     cout << float_result << endl;
28     cout << double_result << endl;
29
30     return 0;
31 }
32
```

Блок-схема до задачі:



Результат виконання програми:

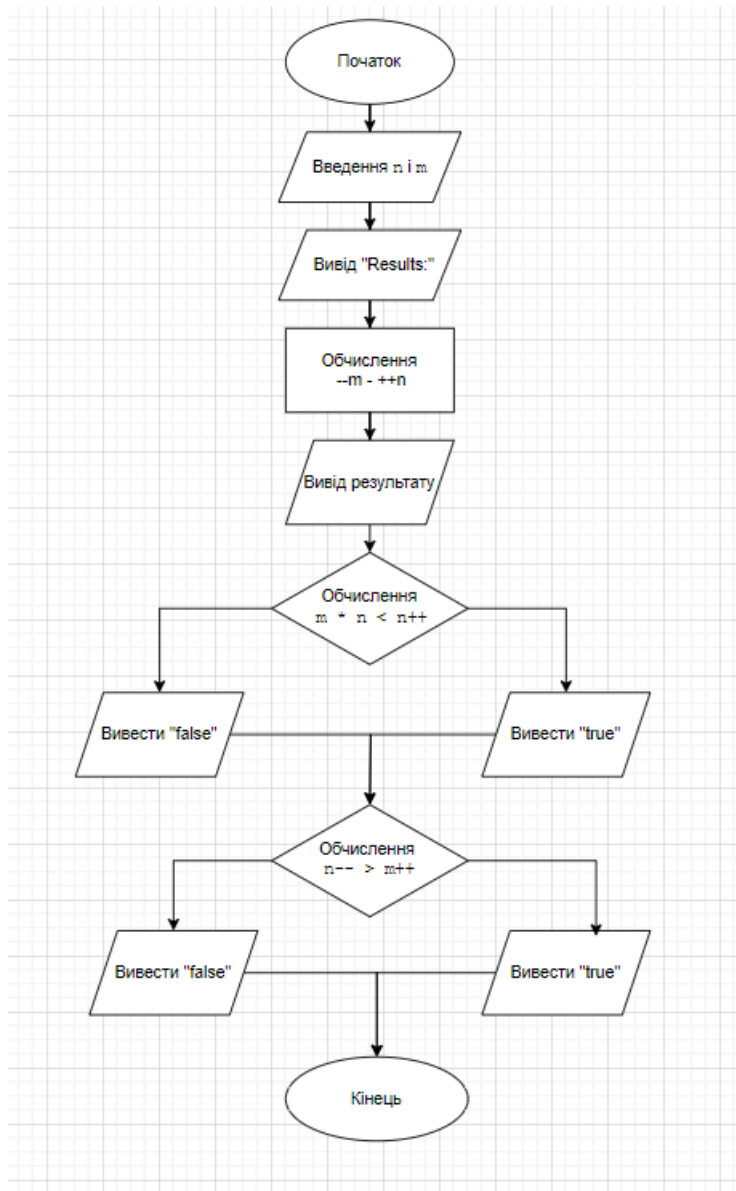
6.25e+06
1.00117

Час виконання завдання ~ 40 хв

2.Розв'язок задачі:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      double n, m;
7
8      cout << "n: ";
9      cin >> n;
10     cout << "m: ";
11     cin >> m;
12
13     cout << "\n Results: \n";
14
15     cout << "- -m - ++n: " << ((-(-m) - (++n)) << endl;
16     cout << "m * n < n++: " << ((m * n < n++) ? "true" : "false") << endl;
17     cout << "n-- > m++: " << ((n-- > m++) ? "true" : "false") << endl;
18
19     return 0;
20 }
```

Блок-схема до задачі:



Результат виконання програми:

```
n: 5
m: 2

Results:
- -m - ++n: -4
m * n < n++: false
n-- > m++: true
```

Час виконання завдання ~ 35 хв

Завдання 2: Algotester Lab 1

Варіант – 2

У вас є стіл, у якого є 4 ніжки, довжини яких вам дано.

Ви хочете зробити ніжки рівної довжини, для цього ви відпиляєте d від кожної ніжки (тобто вам буде дано 4 числа, кожне з яких буде означати відпилювання від відповідної ніжки стола).

Якщо під час відпилювання найдовша ніжка стола буде у 2 рази більша-рівна ніж найменша ніжка - стіл перевернеться, але відпилювати ніжки це вам не завадить.

Тобто якщо $h_{\max} \geq 2 \cdot h_{\min}$ то стіл перевертається. **Увага**, це може статися і між початком та кінцем відпилювання, наприклад коли відпиляють 2, але ще не встигнуть відпиляти 3тю ніжку. Також ми вважаємо що перед відпилюванням стіл не перевернеться.

Ваше завдання сказати чи після усіх маніпуляцій стіл буде цілий та паралельний підлозі. Якщо довжина, яку відріжуть буде більша за довжину ножки - вам треба вивести ERROR.

Увага! Навіть якщо стіл перевернеться - ви все одно відпилюєте ніжки і можете отримати ERROR.

Input

4 цілих числа $h_1, 2, 3, 4$ - довжини ніжок стола 4 цілих числа $d_1, 2, 3, 4$ - довжина, яку відпиляють від відповідної ножки.

Output

YES - якщо стіл буде стояти паралельно площині підлоги та довжина найменшої ніжки не буде рівна нулю.

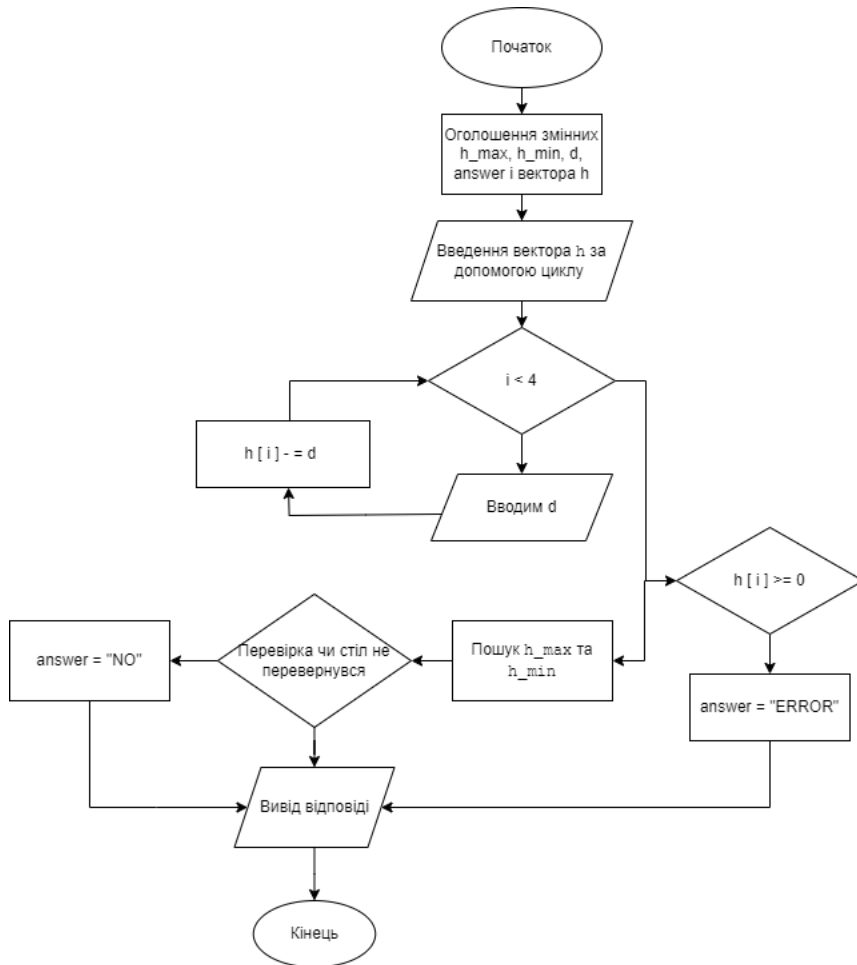
ERROR - у випадку якщо ви відпиляєте більшу довжину ніж має ножка

NO - у інших випадках.

Розв'язок задачі:

```
1  #include <iostream>
2  #include <vector>
3  #include <algorithm>
4
5  using namespace std;
6
7  int main ()
8  {
9      long long h_max, h_min, d;
10
11     vector<long long> h(4);
12
13     string answer = "YES";
14
15     for (int i = 0; i < 4; i++)
16     {
17         cin >> h[i];
18     }
19
20     for (size_t i = 0; i < 4; i++){
21
22         cin >> d;
23         h[i] -= d;
24
25         if ( h[i] >= 0)
26         {
27             h_max = *max_element(h.begin(), h.end());
28             h_min = *min_element(h.begin(), h.end());
29
30             if ( h_max >= h_min * 2 || h_min == 0 )
31             {
32                 answer = "NO";
33             }
34         }
35         else
36         {
37             answer = "ERROR";
38             break;
39         }
40     }
41
42     cout << answer;
43     return 0;
44 }
```

Блок-схема до задачі:



Результат виконання програми:

```
10 10 10 10
1 1 1 1
YES
```

```
10 10 10 10
5 1 1 1
NO
```

```
10 10 10 10
11 1 1 1
ERROR
```

```
2147483648 1 1 1
1 1 1 1
NO
```

```
10 10 10 10
5 5 5 5
NO
```

Час виконання завдання ~ 9 год

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;

Вимоги:

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

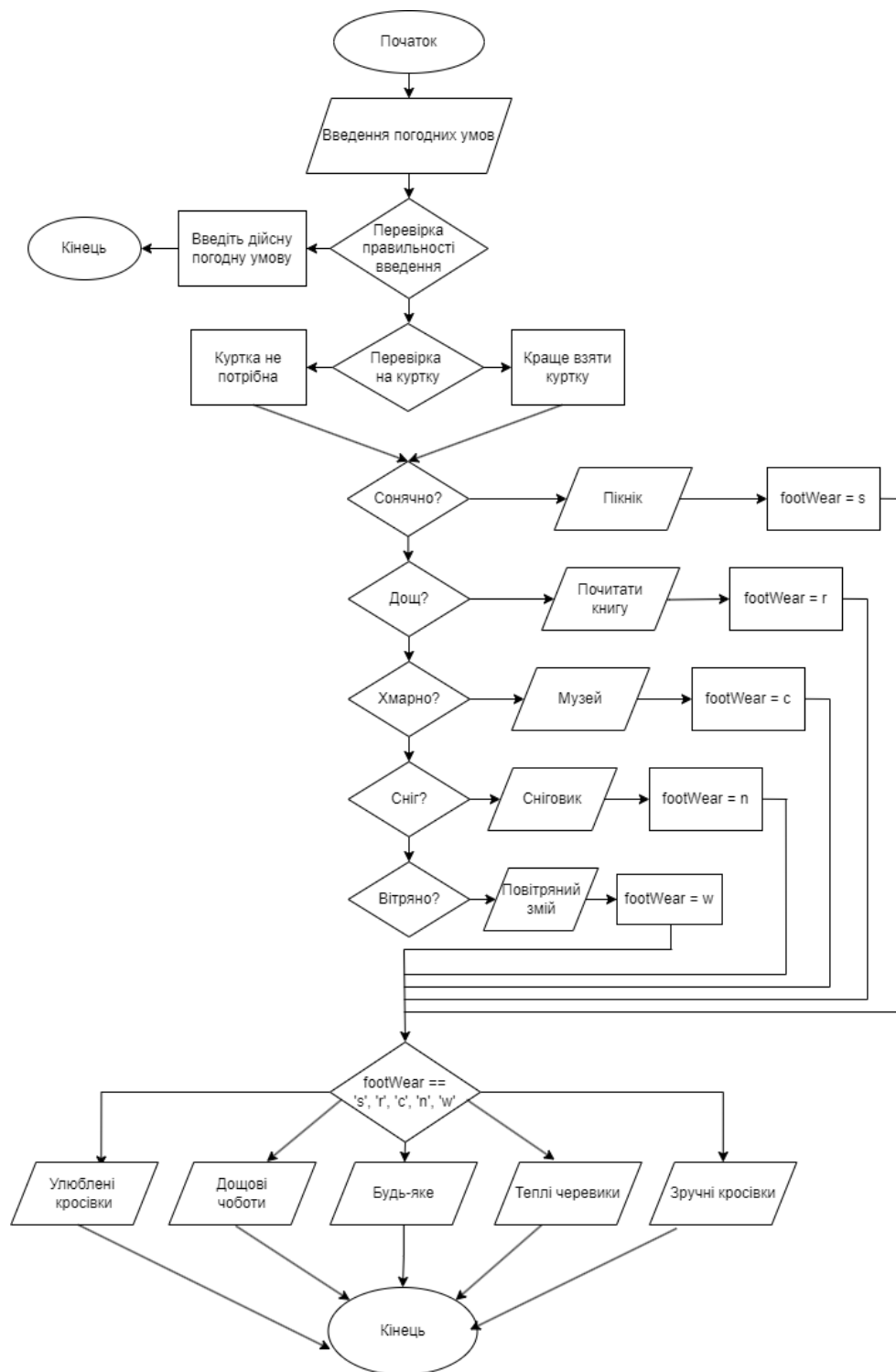
Розв'язок задачі:

```

1  #include <iostream>
2  #include <string>
3
4  using namespace std;
5
6  int main()
7  {
8      string weather;
9      char footWear;
10
11     cout << "Яка зараз погода? (sunny, rainy, cloudy, snowy, windy): ";
12     cin >> weather;
13
14     if (weather != "sunny" && weather != "rainy" && weather != "cloudy" && weather != "snowy" && weather != "windy")
15     {
16         cout << "Будь ласка, введіть дійсну погодну умову!" << endl;
17         return 0;
18     }
19
20     if(weather == "rainy" || weather == "snowy")
21     {
22         cout << "Сьогодні краще взяти куртку!" << endl;
23     }
24     else
25     {
26         cout << "Куртка не потрібна" << endl;
27     }
28
29     if (weather == "sunny")
30     {
31         cout << "Чудовий день для пікніка!" << endl;
32         footWear = 's';
33     }
34     else if (weather == "rainy")
35     {
36         cout << "Чудова погода, щоб почитати книгу всередині" << endl;
37         footWear = 'r';
38     }
39     else if (weather == "cloudy")
40     {
41         cout << "Може відвідати музей?" << endl;
42         footWear = 'c';
43     }
44     else if (weather == "snowy")
45     {
46         cout << "Як щодо того, щоб зліпити сніговика" << endl;
47         footWear = 'n';
48     }
49     else if (weather == "windy")
50     {
51         cout << "Запустіть повітряного змія, якщо він у вас є!" << endl;
52         footWear = 'w';
53     }
54
55     switch (footWear)
56     {
57     case 's':
58         cout << "Взуй улюблені кросівки!" << endl;
59         break;
60
61     case 'r':
62         cout << "Дощові чоботи - гарна ідея!" << endl;
63         break;
64
65     case 'c':
66         cout << "Сьогодні підходить будь-яке взуття" << endl;
67         break;
68
69     case 'n':
70         cout << "Взуй теплі черевики!" << endl;
71         break;
72
73     case 'w':
74         cout << "Вибери зручні кросівки" << endl;
75         break;
76
77     default:
78         cout << "Не знаю, яке взуття обрати(";
79         break;
80     }
81
82     return 0;
83 }

```


Блок-схема до задачі:



Результат виконання програми:

```

яка зараз погода? (sunny, rainy, cloudy, snowy, windy): windy
Куртка не потрібна
Запустіть повітряного змія, якщо він у вас є!
Вибери зручні кросівки
  
```

Час виконання завдання ~ 40 хв

Завдання 4: Self Practice Algotester Task 1

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

Тож уявіть 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 nn 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?

Input

У єдиному рядку задано одне натуральне число nn — вартість подарунку.

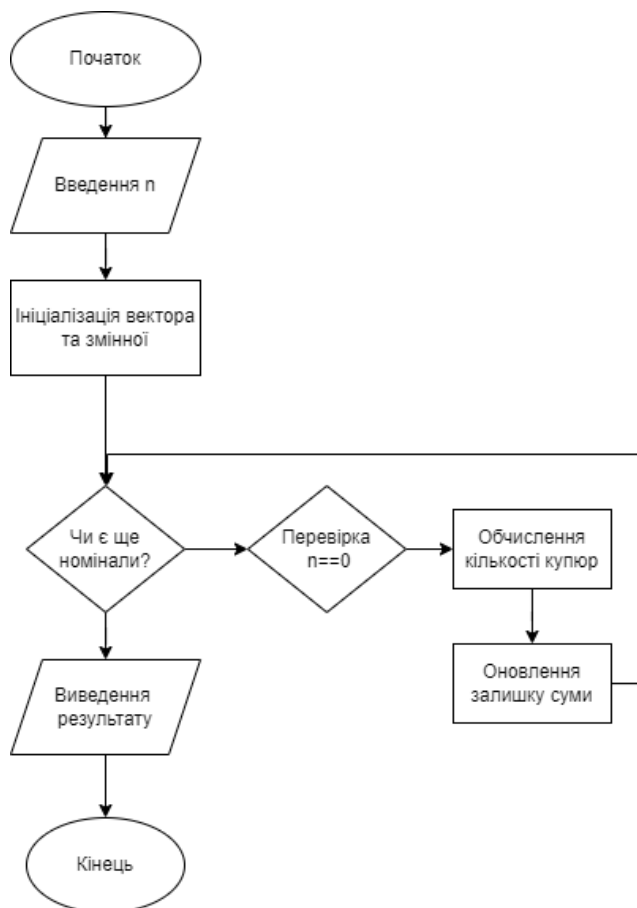
Output

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

Розв'язок задачі:

```
1  #include <iostream>
2  #include <vector>
3
4  using namespace std;
5
6  int main() {
7      int n;
8      cin >> n;
9
10     vector<int> denominations = {1000, 500, 200, 100, 50, 20, 10, 5, 2, 1};
11     int count = 0;
12
13     for (int denom : denominations) {
14         if (n == 0)
15             break;
16         count += n / denom;
17         n %= denom;
18     }
19
20     cout << count << endl;
21
22     return 0;
23 }
24
```

Блок-схема до задачі:



Результат виконання програми:

1773
7

Час виконання завдання ~ 20 хв

Взаємодія з командою:

IN PROGRESS							
6 ... + Add Task							
Name	Assignee	Due date	Priority	Status	Comments		
<div><div></div>Alina Khodatska - Epic 1</div> <div>9/14</div>	<div>A</div>	<div></div>	<div><div></div>High</div>	<div><div></div>IN PROGRE...</div>	<div></div>	<div>...</div>	
<div><div></div>Tofan Max - Epic 1</div> <div>0/14</div>	<div>F</div>	<div></div>	<div><div></div>High</div>	<div><div></div>IN PROGRE...</div>	<div></div>	<div>...</div>	
<div><div></div>Nazar Nedosika - Epic 2</div> <div>4/5</div>	<div>NN</div>	<div></div>	<div><div></div>High</div>	<div><div></div>IN PROGRE...</div>	<div></div>	<div>...</div>	
<div><div></div>Tofan Max - Epic 2</div> <div>0/5</div>	<div>F</div>	<div></div>	<div><div></div>High</div>	<div><div></div>IN PROGRE...</div>	<div></div>	<div>...</div>	
<div><div></div>Kobzar Artem - Epic 2</div> <div>4/5</div>	<div>KC</div>	<div></div>	<div><div></div>High</div>	<div><div></div>IN PROGRE...</div>	<div></div>	<div>...</div>	
<div><div></div>Alina Khodatska - Epic 2</div> <div>0/5</div>	<div>A</div>	<div></div>	<div><div></div>High</div>	<div><div></div>IN PROGRE...</div>	<div></div>	<div>...</div>	
<div>+ Add Task</div>							

Nazar Nedosika - Epic 2

Checklists 4/5

Checklist (4/5)

- ☒ VNS-Lab 1 Tasks 1,2
- ☒ Algotester-Lab 1 Variant 1
- ☒ Practice Task
- ☒ Self Practice Tasks

☐ Git Branch, PR

+ New checklist item



Висновок: В цьому епіку я навчився користуватися оператором if-else та switch. Навчився малювати блок-схеми до задач, розглянув цикли та зробив додаткові задачі, щоб закріпити знання.