

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



Звіт

про виконання лабораторних та практичних робіт блоку № 3
На тему: «Цикли. Вкладені Цикли. Завершення виконання циклів.
Функції. Простір імен. Перевантаження функцій. Функції з змінною
кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.»
з дисципліни: «Основи програмування»

до:

ВНС Лабораторної Роботи № 2
ВНС Лабораторної Роботи № 3
ВНС Лабораторної Роботи № 7
Практичних Робіт до блоку № 3

Виконав:
Студент групи ШІ-13
Бойко Роман Андрійович

Львів 2024

Тема роботи: Цикли. Вкладені Цикли. Завершення виконання циклів. Функції. Простір імен. Перевантаження функцій. Функції з змінною кількістю параметрів (еліпсис). Рекурсія. Вбудовані функції.

Мета роботи: Навчитися використовувати функції різних видів, цикли та виконувати завдання над ними. Також будувати блок-схеми складніших типів.

Теоретичні відомості:

- цикл for
- цикл while
- цикл do, while
- функції

Джерела:

- [C++ • Теорія • Урок 32 • while](#)
- [C++ • Теорія • Урок 33 • do while](#)
- [C++ • Теорія • Урок 34 • for](#)
- [C++ • Теорія • Урок 35 • for по діапазону](#)
- [C++ • Теорія • Урок 38 • Вкладені конструкції](#)
- [C++ • Теорія • Урок 46 • Параметри за замовчуванням](#)
- [C++ • Теорія • Урок 48 • Перезавантаження функцій](#)

Виконання роботи

Особистий варіант - 1

Завдання 1: Practice task

Ви створюєте просту програму керування бібліотекою. Книги в бібліотеці є, користувачі можуть їх взяти або повернути.

Програма повинна вміти

- Перерахувати всі книги.
- Дозволити взяти книгу (за наявності).
- Дозволити повернення книги.

Структури даних

- Використовуйте масив або вектор для зберігання назв книг.
- Використовуйте інший масив або вектор для збереження стану доступності кожної книги.

Мета Задачі

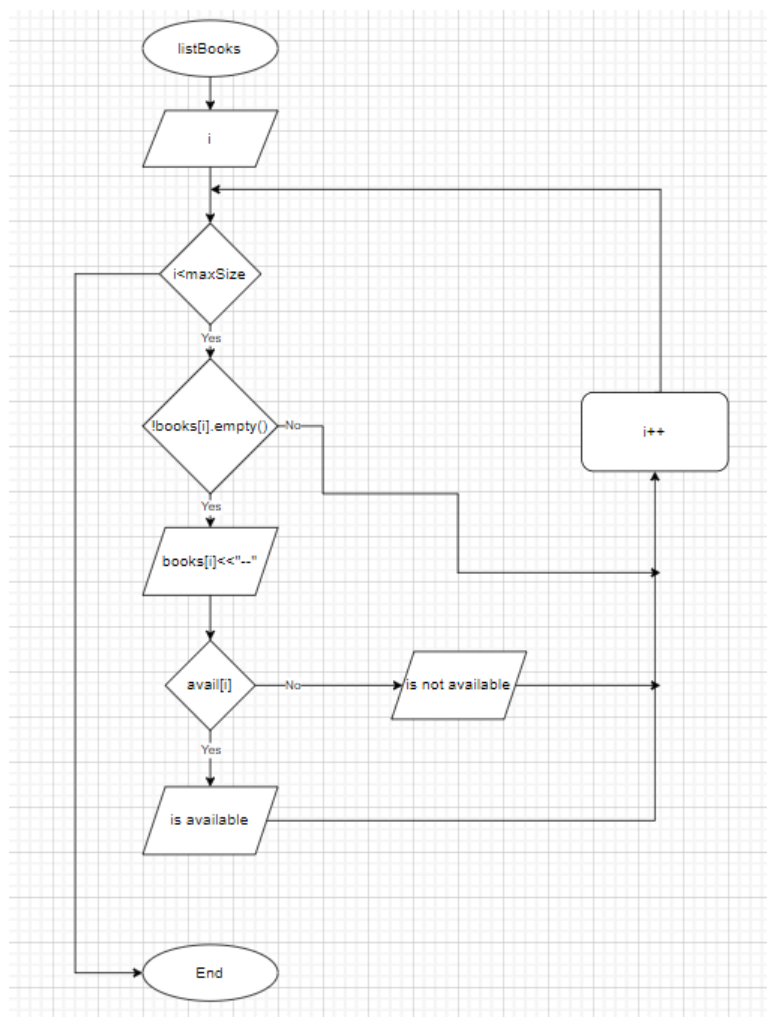
Навчитися користуватися операторами циклів та функцією переходу на мітку:

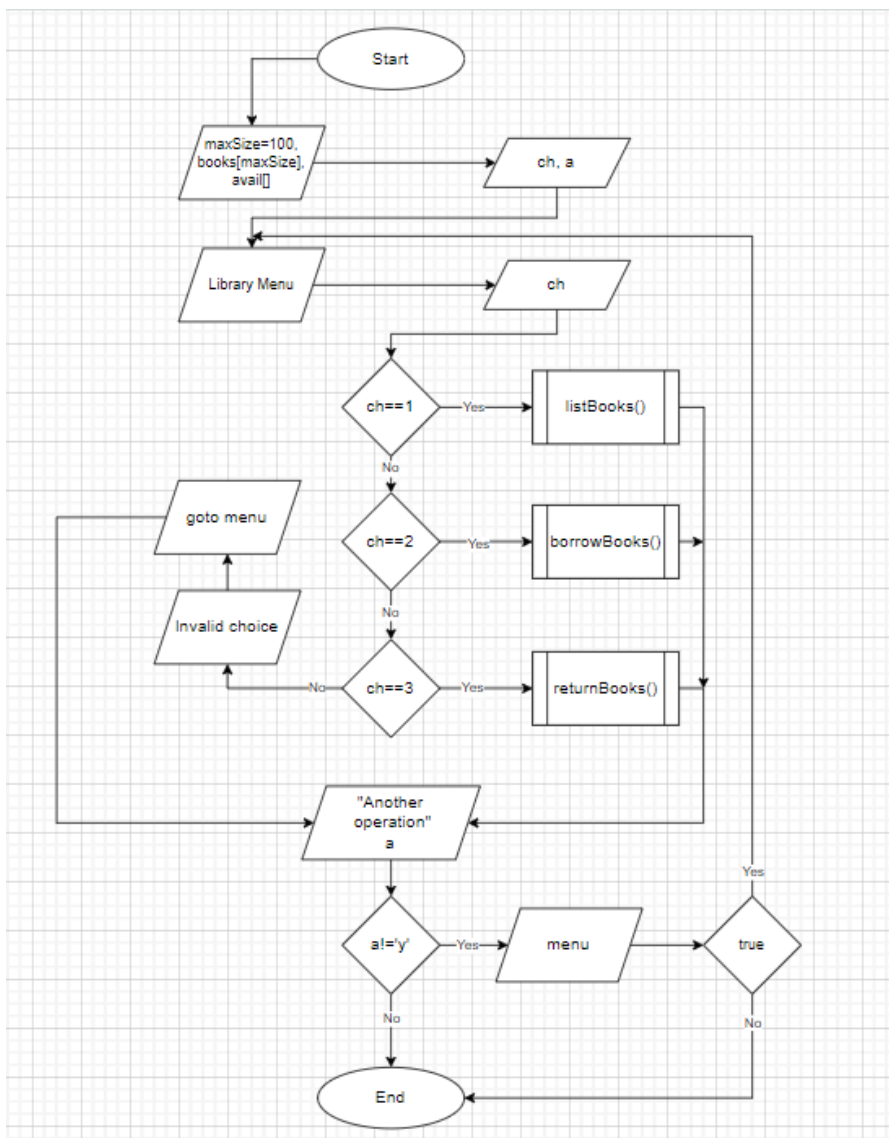
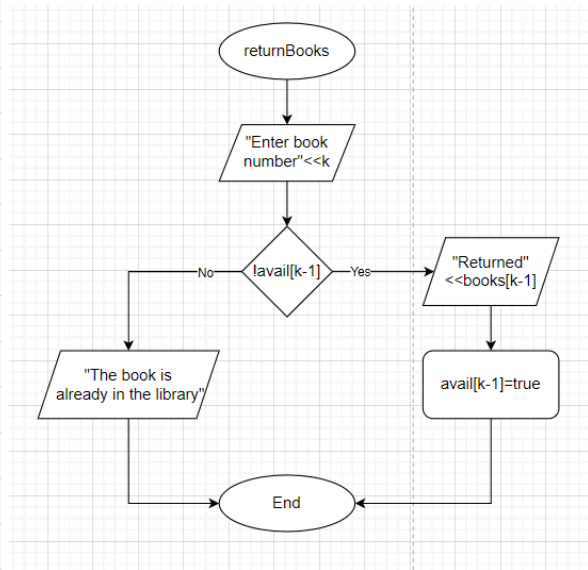
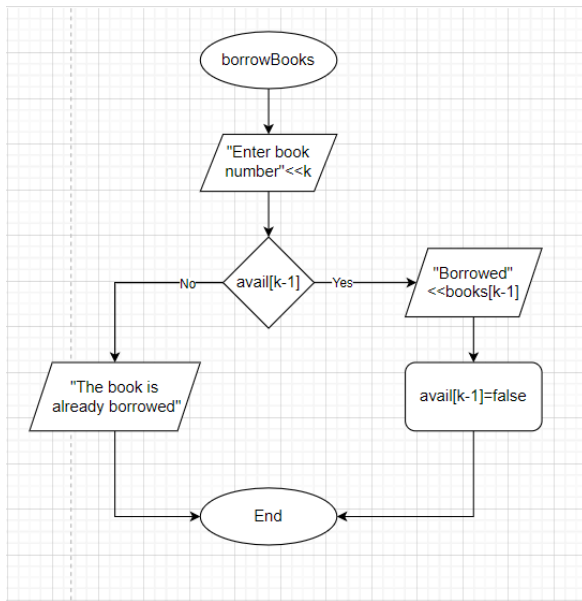
1. `for () { ... }`
2. `for each`
3. `while () { ... }`
4. `do { ... } while ()`
5. `go to`

Вимоги:

1. `while`: продовжувати працювати, доки користувач не вирішить вийти.
2. `do while`: Після кожної операції (позичити, повернути, перерахувати) запитуйте користувача, чи хоче він виконати іншу операцію. Якщо так, поверніться назад.
3. `for`: список усіх книг за допомогою циклу.
4. `for each`: перевірити наявність кожної книги.
5. `goto`: якщо користувач вводить неправильний вибір, використовуйте `goto`, щоб перенаправити його до головного меню.

Блок-схема до задачі:





Розв'язок задачі:

```
1  #include <iostream>
2  #include <string>
3  #include "stdio.h"
4  using namespace std;
5  const int maxSize = 100;
6  string books[maxSize] = {"Harry Potter",
7                           "To Kill a Mockingbird",
8                           "Pride and Prejudice",
9                           "1984",
10                          "Jane Eyre",
11                          "The Catcher in the Rye",
12                          "The Lord of the Rings"};
13
14  bool avail[] = {true, true, true, true, true, true, true};
15  void listBooks();
16  void borrowBooks();
17  void returnBooks();
18  int main()
19  {
20
21      int ch;
22      do
23      {
24          char a;
25          cout << "Library Menu\n";
26          cout << "1. List all books\n";
27          cout << "2. Borrow a book\n";
28          cout << "3. Return a book\n";
29          cout << "Enter your choice: ";
30          cin >> ch;
31
32          if (ch == 1)
33          {
34              listBooks();
35          }
36          else if (ch == 2)
37          {
38              borrowBooks();
39          }
40          else if (ch == 3)
41          {
42              returnBooks();
43          }
44          else
45          {
46              cout << "Invalid choice!" << endl;
47              goto menu;
48          }
49
50          cout << "Do you want to choose another operation? (y/n): ";
51          cin >> a;
52          if (a != 'y')
53          {
54              break;
55          }
56          menu:
57          while (true);
58      }
59
60  void listBooks()
61  {
```

```

61     for (int i = 0; i < maxSize; i++)
62     {
63         if (!books[i].empty())
64         {
65             cout << "Book " << i + 1 << ": " << books[i] << " -- ";
66             if (avail[i])
67             {
68                 cout << " is available!" << endl;
69                 continue;
70             }
71             else
72             {
73                 cout << " is not available" << endl;
74                 continue;
75             }
76         }
77     }
78 }
79
80 void borrowBooks()
81 {
82     int k;
83     cout << "Enter book number you want to borrow: ";
84     cin >> k;
85     if (avail[k - 1])
86     {
87         cout << "You borrowed the book: " << books[k - 1] << endl;
88         avail[k - 1] = false;
89     }
90     else

```

```

91     {
92         cout << "The book is already borrowed!" << endl;
93     }
94 }
95
96 void returnBooks()
97 {
98     int k;
99     cout << "Enter book number you want to return: ";
100    cin >> k;
101    if (!avail[k - 1])
102    {
103        cout << "You returned the book: " << books[k - 1] << endl;
104        avail[k - 1] = true;
105    }
106    else
107    {
108        cout << "The book is already in the library!" << endl;
109    }
110 }

```

Вивід в терміналі:

```

Library Menu
1. List all books
2. Borrow a book
3. Return a book
Enter your choice: 1
Book 1: Harry Potter --  is available!
Book 2: To Kill a Mockingbird --  is available!
Book 3: Pride and Prejudice --  is available!
Book 4: 1984 --  is available!
Book 5: Jane Eyre --  is available!
Book 6: The Catcher in the Rye --  is available!
Book 7: The Lord of the Rings --  is available!
Do you want to choose another operation? (y/n): █

```

```

Do you want to choose another operation? (y/n): y
Library Menu
1. List all books
2. Borrow a book
3. Return a book
Enter your choice: 2
Enter book number you want to borrow: 1
You borrowed the book: Harry Potter
Do you want to choose another operation? (y/n): y
Library Menu
1. List all books
2. Borrow a book
3. Return a book
Enter your choice: 3
Enter book number you want to return: 5
The book is already in the library!

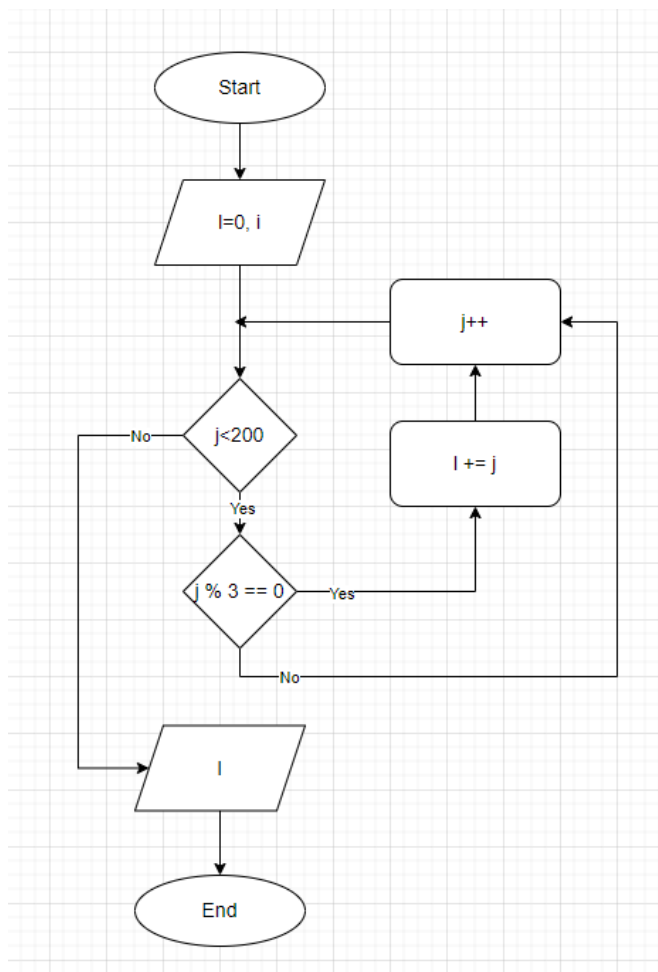
```

Час виконання завдання: ~ 25 хвилин

Завдання 2: VNS Lab 2

Знайти суму цілих додатніх чисел, кратних 3 і менших 200.

Блок-схема до задачі:



Розв'язок задачі:

```
1  #include <iostream>
2
3  using namespace std;
4
5  int main()
6  {
7      int l = 0;
8      for (int j = 0; j < 200; j++)
9      {
10         if (j % 3 == 0)
11         {
12             l += j;
13         }
14     }
15     cout << l;
16 }
```

Вивід в терміналі:

6633

Час виконання завдання: ~ 5 хвилин

Завдання 3: VNS Lab 3

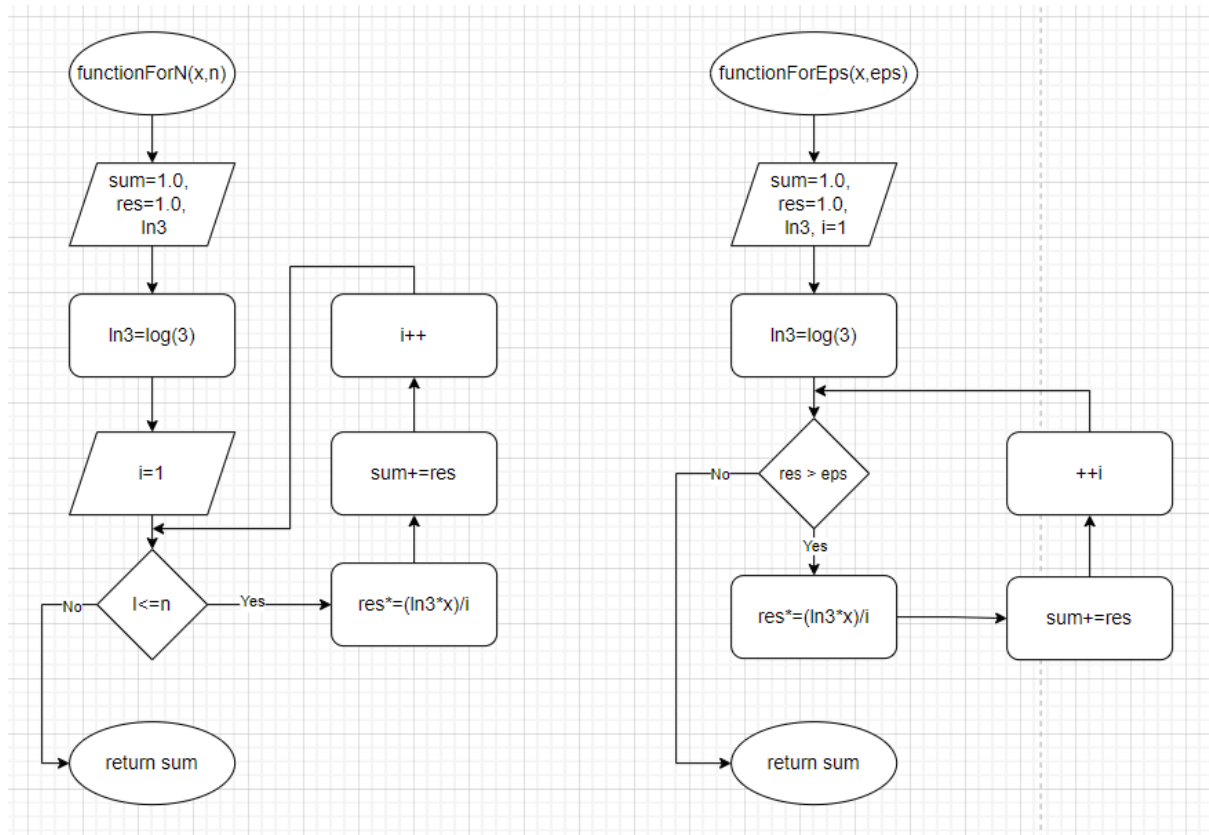
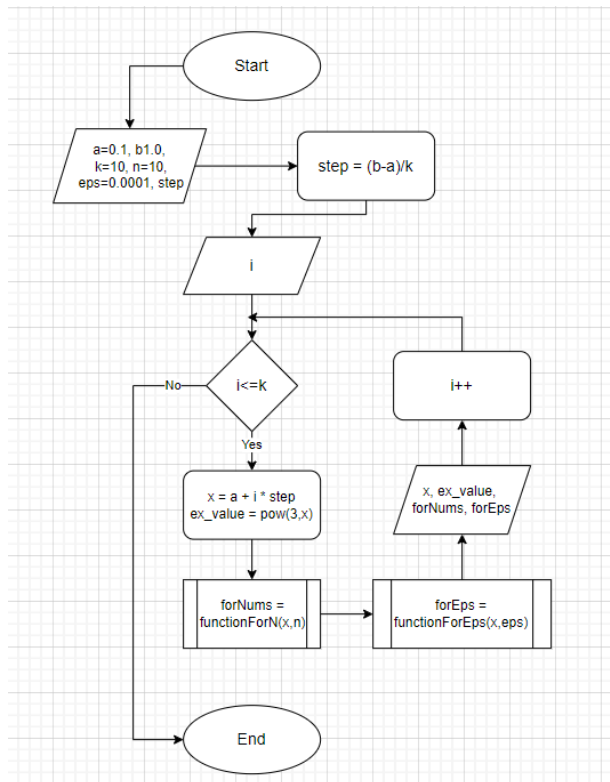
Для x , що змінюється від a до b з кроком $(b-a)/k$, де $(k=10)$, обчислити функцію $f(x)$, використовуючи її розклад в степеневий ряд у двох випадках:

- а) для заданого n ;
- б) для заданої точності ϵ ($\epsilon=0.0001$).

Для порівняння знайти точне значення функції.

функція	діапазон зміни аргумент y	n	сума
$y = 3^x$	$0,1 \leq x \leq 1$	10	$S = 1 + \frac{\ln 3}{1!}x + \frac{\ln^2 3}{2!}x^2 + \dots + \frac{\ln^n 3}{n!}x^n$

Блок-схема до задачі:



Розв'язок задачі:

```
1  #include <iostream>
2  #include <iomanip>
3  #include <cmath>
4  using namespace std;
5  double functionForN(double x, int n);
6  double functionForEps(double x, double eps);
7  int main()
8  {
9      double a = 0.1;
10     double b = 1.0;
11     int k = 10, n = 10;
12     double eps = 0.0001;
13     double step = (b - a) / k;
14
15     for (int i = 0; i <= k; i++)
16     {
17         double x = a + i * step;
18         double ex_value = pow(3, x);
19         double forNums = functionForN(x, n);
20         double forEps = functionForEps(x, eps);
21         cout << "x = " << setprecision(4) << x << endl;
22         cout << "res = " << setprecision(8) << ex_value << endl;
23         cout << "res in n = " << setprecision(8) << forNums << endl;
24         cout << "res in eps = " << setprecision(8) << forEps << endl;
25         cout << endl;
26     }
27 }
```

```
28 double functionForN(double x, int n)
29 {
30     double sum = 1.0, res = 1.0;
31     double ln3 = log(3);
32     for (int i = 1; i <= n; i++)
33     {
34         res *= (ln3 * x) / i;
35         sum += res;
36     }
37     return sum;
38 }
39 double functionForEps(double x, double eps)
40 {
41     double sum = 1.0, res = 1.0;
42     double ln3 = log(3);
43     int i = 1;
44     while (res > eps)
45     {
46         res *= (ln3 * x) / i;
47         sum += res;
48         ++i;
49     }
50     return sum;
51 }
```

Вивід в терміналі:

```
x = 0.1  
res = 1.1161232  
res in n = 1.1161232  
res in eps = 1.116123
```

```
x = 0.19  
res = 1.2321201  
res in n = 1.2321201  
res in eps = 1.2321167
```

```
x = 0.28  
res = 1.3601724  
res in n = 1.3601724  
res in eps = 1.3601711
```

```
x = 0.37  
res = 1.5015329  
res in n = 1.5015329  
res in eps = 1.5015263
```

```
x = 0.46  
res = 1.6575849  
res in n = 1.6575849  
res in eps = 1.6575831
```

```
x = 0.55  
res = 1.8298551  
res in n = 1.8298551  
res in eps = 1.8298487
```

```
x = 0.64  
res = 2.020029  
res in n = 2.020029  
res in eps = 2.0200274
```

```
x = 0.73  
res = 2.2299674  
res in n = 2.2299674  
res in eps = 2.2299628
```

```
x = 0.82  
res = 2.4617244  
res in n = 2.4617244  
res in eps = 2.4617125
```

```
x = 0.91  
res = 2.7175675  
res in n = 2.7175675  
res in eps = 2.7175645
```

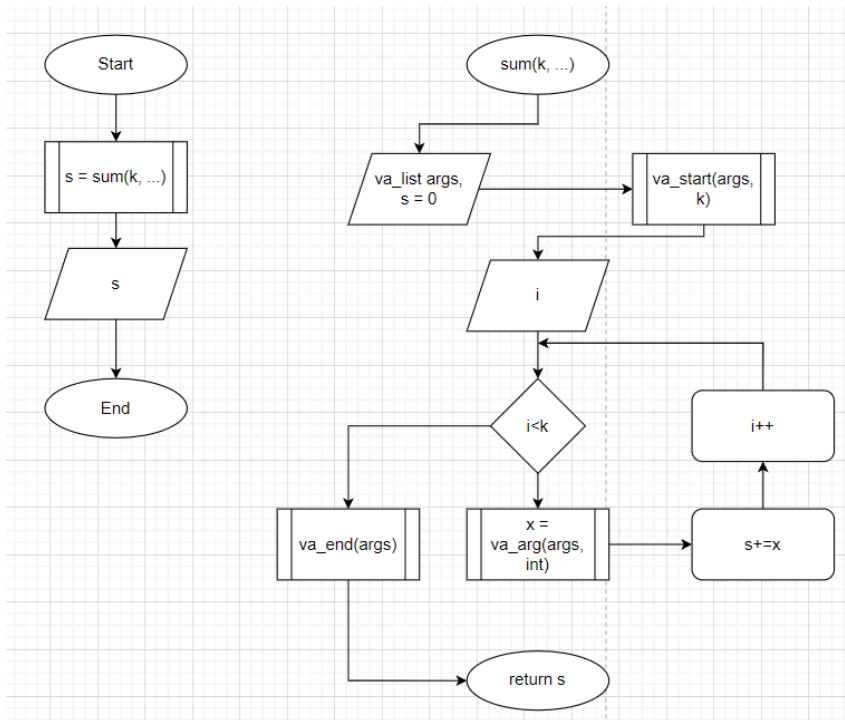
```
x = 1  
res = 3  
res in n = 2.9999999  
res in eps = 2.9999928
```

Час виконання завдання: ~ 45 хвилин

Завдання 4: VNS Lab 7_1

Написати функцію `sum` зі змінною кількістю параметрів, що знаходить суму чисел типу `int`. Написати викликаючу функцію `main`, що звертається до функції `sum` не менш трьох разів з кількістю параметрів 3, 7, 11.

Блок-схема до задачі:



Розв'язок задачі:

```

1  #include <iostream>
2  #include <stdarg.h>
3  using namespace std;
4  int sum(int k, ...);
5  int main()
6  {
7
8      int s = sum(3, 1, 2, 7);
9      cout << s;
10 }
11
12 int sum(int k, ...)
13 {
14     va_list args;
15     va_start(args, k);
16     int s = 0;
17     for (int i = 0; i < k; i++)
18     {
19         int x = va_arg(args, int);
20         s += x;
21     }
22     va_end(args);
23     return s;
24 }
25

```

Вивід в терміналі:

10

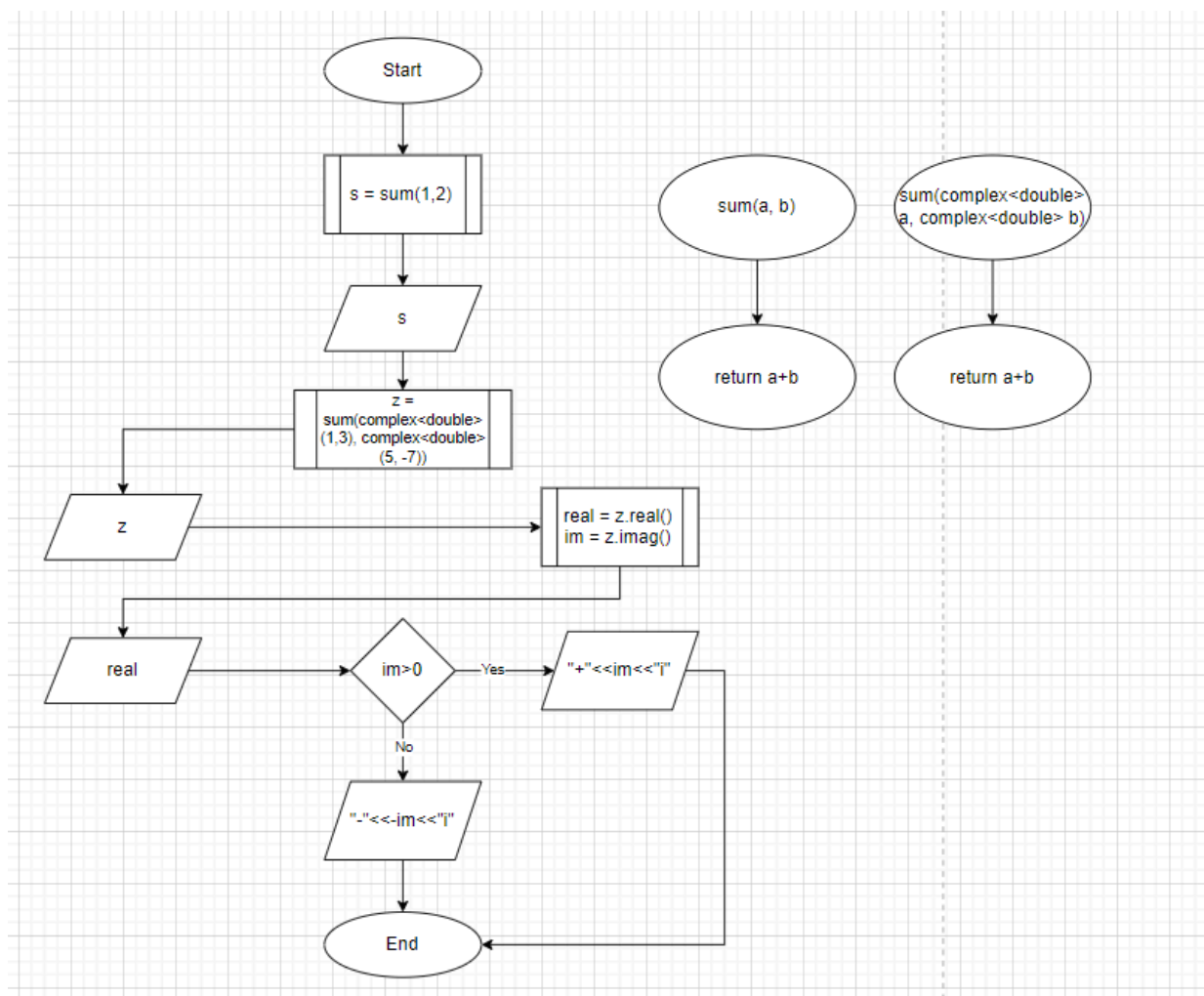
Час виконання завдання: ~ 30 хвилин

Завдання 4: VNS Lab 7_2

Написати перевантажені функції й основну програму, що їх викликає.

- а) для додавання цілих чисел;
- б) для додавання комплексних чисел.

Блок-схема до задачі:



Розв'язок задачі:

```
1  #include <iostream>
2  #include <complex>
3  using namespace std;
4  int sum(int a, int b)
5  {
6      return a + b;
7  }
8  complex<double> sum(complex<double> a, complex<double> b)
9  {
10     return a + b;
11 }
12 int main()
13 {
14
15     int s = sum(1, 2);
16     cout << s;
17     cout << endl;
18     complex<double> z = sum(complex<double>(1, 3), complex<double>(5, -7));
19     double real = z.real();
20     double im = z.imag();
21     cout << real;
22     if (im > 0)
23     {
24         cout << "+" << im << "i";
25     }
26     else
27     {
28         cout << "-" << -im << "i";
29     }
30 }
31
```

Вивід в терміналі:

```
3
6-4i
```

Час виконання завдання: ~ 25 хвилин

Завдання 5: Self Practice Task 1

Депутатські гроші

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 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.

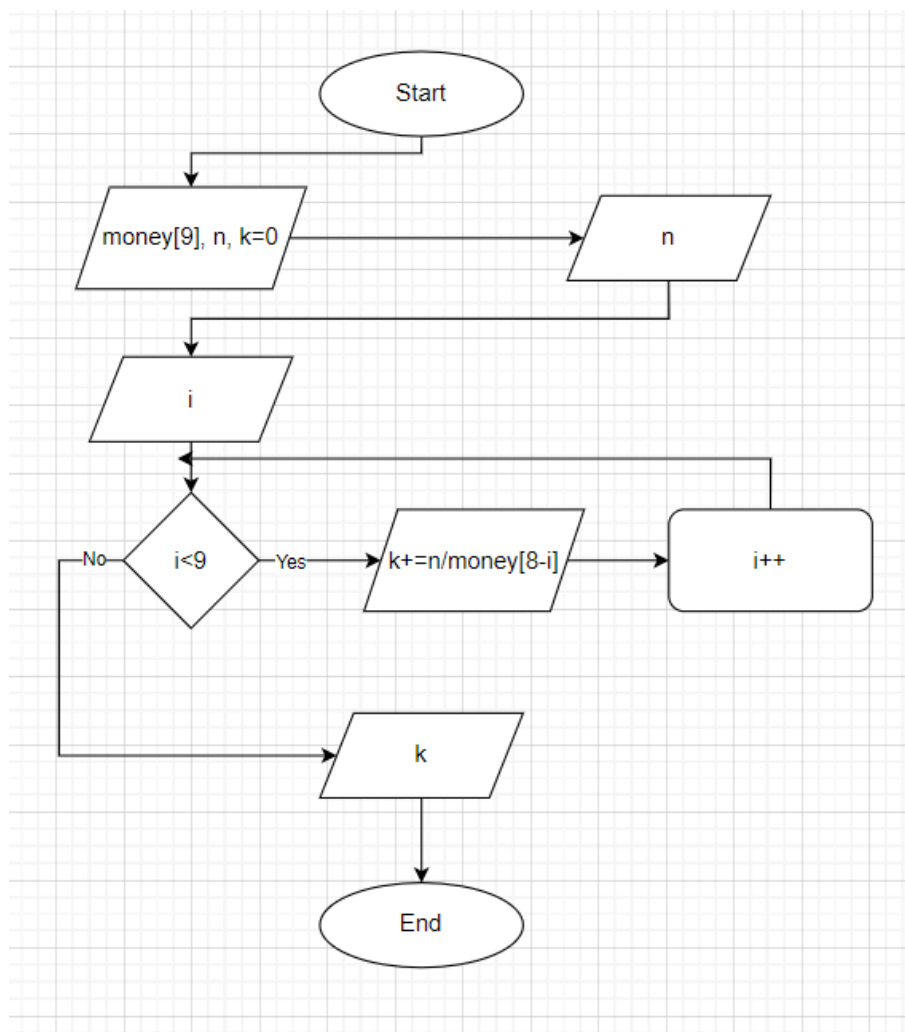
Вихідні дані

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

Обмеження

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

Блок-схема до задачі:



Розв'язок задачі:

```
1  #include <iostream>
2  using namespace std;
3
4  int main()
5  {
6      int money[9] = {1, 2, 5, 10, 20, 50, 100, 200, 500};
7      long int n;
8      int k = 0;
9      cin >> n;
10     for (int i = 0; i < 9; i++)
11     {
12         k += n / money[8 - i];
13         n = n % money[8 - i];
14     }
15     cout << k;
16 }
```

Вивід в терміналі:

```
567
5
```

Час виконання завдання: ~ 20 хвилин

Завдання 6: Self Practice Task 2

Про фальсифікації, перерахунок та перевибори

І ось відбулися перевибори, голоси перераховано.

Цього разу у виборах брали участь n політичних партій. Для кожної партії відомо, скільки голосів вона набрала на перших виборах і скільки — на перевиборах. Тепер голові ЦВК потрібно відповідати за різницю цих результатів по кожній партії — а саме, якщо кількість голосів якоїсь партії на виборах відрізняється від кількості голосів на перевиборах на d , то від його зарплати віднімуть d мільярдів доларів. Поточна зарплата голови — s мільярдів доларів.

Визначте його зарплату (у мільярдах доларів) після виборів. Зауважте, що зарплата голови може стати від'ємною.

Вхідні дані

У першому рядку задано два натуральні числа n та s — кількість партій та зарплату голови ЦВК.

У другому рядку задано n невід'ємних чисел — кількість голосів відповідної партії на виборах.

У третьому рядку задано аналогічно результати перевиборів.

Вихідні дані

У єдиному рядку виведіть ціле число — зарплату голови після виборів (у мільярдах, звісно).

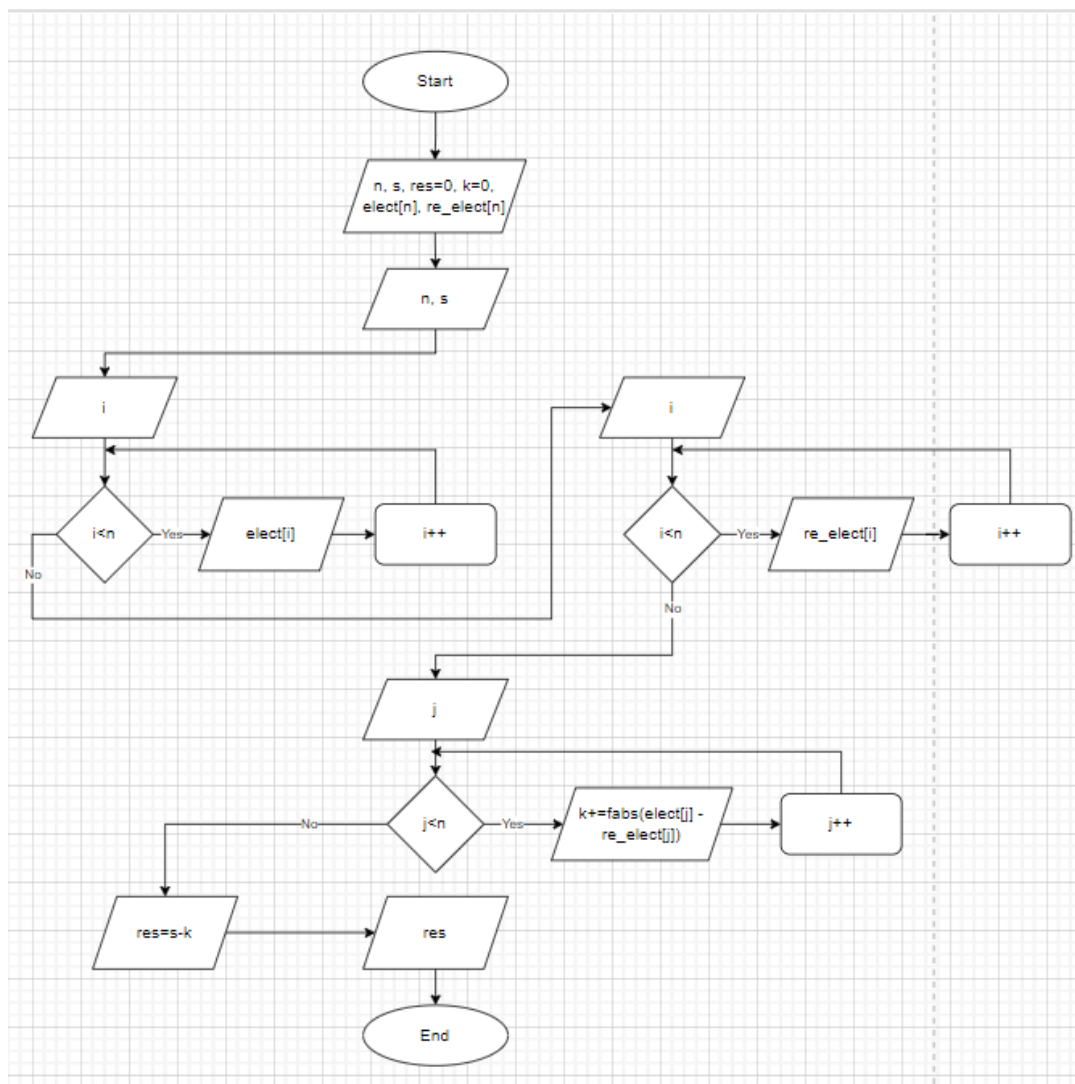
Обмеження

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

$$1 \leq s \leq 10^9,$$

кількість голосів кожної партії не перевищує 10^4

Блок-схема до задачі:



Розв'язок задачі:

```
1  #include <iostream>
2  #include <cmath>
3  using namespace std;
4  int main()
5  {
6      int n, s;
7      int res = 0, k = 0;
8      cin >> n >> s;
9      int elect[n] = {};
10     int re_elect[n] = {};
11     for (int i = 0; i < n; i++)
12     {
13         cin >> elect[i];
14     }
15     for (int i = 0; i < n; i++)
16     {
17         cin >> re_elect[i];
18     }
19     for (int j = 0; j < n; j++)
20     {
21         k += fabs(elect[j] - re_elect[j]);
22     }
23     res = s - k;
24     cout << res;
25 }
```

Вивід в терміналі:

```
4 7
3 4 5 7
4 5 1 7
1
```

Час виконання завдання: ~ 15 хвилин

Висновок:

В цьому епіку я навчився користуватися циклами for, while, do while, функціями різних видів, також розібрав бібліотеку cstdarg для задачі з невідомою кількістю параметрів.