**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ**

**ІМЕНІ ТАРАСА ШЕВЧЕНКА**

**ФАКУЛЬТЕТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ**

**Кафедра інтелектуальних технологій**

Лабораторна робота № 5

з дисципліни «Функції мови С/С++»

**Виконав студент**

**групи КН-12**

**Сімак О. С.**

**Перевірив**

**Круглов О. І.**

**Київ – 2020**

**Завдання 1**

Описати функцію sumRange(A, B) цілого типу, що знаходить суму всіх цілих чисел від A до B включно (A і B – цілі). Якщо A>B, то функція повертає 0. За допомогою цієї функції знайдіть суми чисел від X до Y і від Y до Z (X, Y, Z задає користувач).

1. **Математична постановка задачі (МПЗ).**

*Вхідні дані*: A, B–змінні цілого типу.

*Вихідні дані*: sum - результати розрахунку цілого типу.

*Математична модель задачі*: sum =

Подання МПЗ у вигляді таблиці:

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| A, B –змінні цілого типу. | sum = 0;  i = A, B: sum += i | sum - результати розрахунку цілого типу. |

Якщо подати функцію як “чорну скриню”, маємо:



Рисунок 1 – Чорна скриня функції задачі 1.

Якщо подати головну програму як “чорну скриню”, маємо:



Рисунок 2 – Чорна скриня головної функції задачі 1.

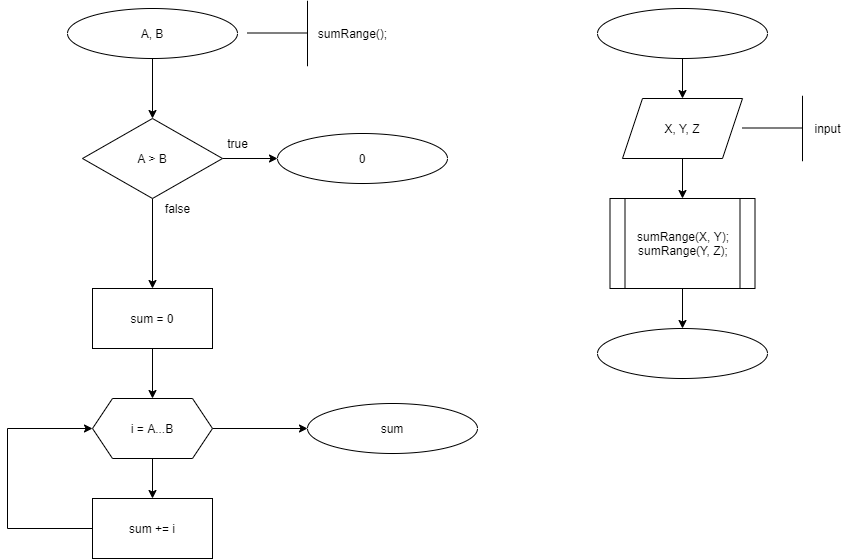
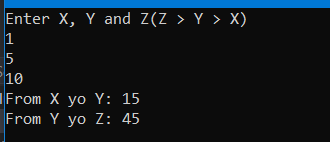


Рисунок 3 - Схема алгоритму задачі 1.

1. **Тестові приклади.**

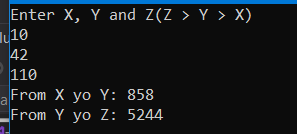
Вхідні дані: X = 1, Y = 5, Z = 10;

Вихідні дані: 15, 45



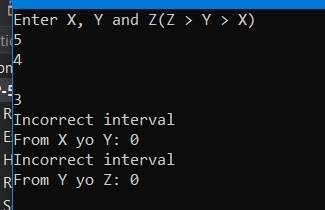
Вхідні дані: X = 10, Y = 42, Z = 110;

Вихідні дані: 858, 5244



Вхідні дані: X = 5, Y = 4, Z = 3;

Вихідні дані: 0, 0(некоректні дані)



**3. Текст програми:**

#include <iostream>

using namespace std;

int sunRange(int A, int B) {

if (A > B) {

cout << "Incorrect interval" << endl;

return 0;

}

int sum = 0;

for (int i = A; i <= B; i++) {

sum += i;

}

return sum;

}

int main()

{

cout << "Enter X, Y and Z(Z > Y > X)" << endl;

int X, Y, Z;

cin >> X >> Y >> Z;

cout << "From X yo Y: " << sunRange(X, Y) << endl;

cout << "From Y yo Z: " << sunRange(Y, Z) << endl;

return 0;

}

**Завдання 2**

Описати функцію іsSquare (K) логічного типу, яка повертає true, якщо цілий параметр K (К>0) є квадратом деякого цілого числа, і false у протилежному випадку. За допомогою цієї функції знайти кількість квадратів у наборі з 10 цілих додатних чисел.

1. **Математична постановка задачі (МПЗ).**

*Вхідні дані*: K –змінна логічного типу.

*Вихідні дані*: isSquare - результати логічного типу.

*Математична модель задачі*: Якщо ціле, тоді true, інакше false

Подання МПЗ у вигляді таблиці:

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| K –змінна логічного типу. | Якщо ціле, тоді true, інакше false | isSquare - результати логічного типу. |

Якщо подати функцію як “чорну скриню”, маємо:



Рисунок 4 – Чорна скриня функції задачі 2.

Якщо подати головну програму як “чорну скриню”, маємо:



Рисунок 5 – Чорна скриня головної функції задачі 2.

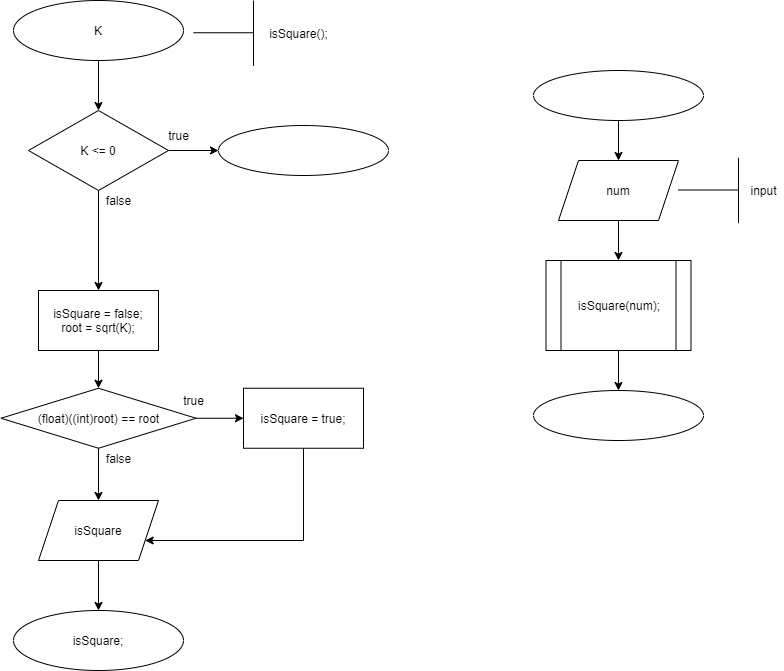
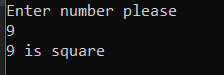


Рисунок 6 - Схема алгоритму задачі 2.

1. **Тестові приклади.**

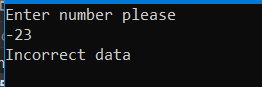
Вхідні дані: num = 9;

Вихідні дані: true;



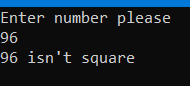
Вхідні дані: num = -23;

Вихідні дані: incorrect data;



Вхідні дані: num = 96;

Вихідні дані: false;



**3. Текст програми:**

#include <iostream>

#include <cmath>

using namespace std;

bool isSquare(int K) {

if (K <= 0) {

cout << "Incorrect data" << endl;

return 0;

}

bool isSquare = false;

float root = sqrt(K);

if ((float)((int)root) == root) { //проверка на целое число

isSquare = true;

cout << K << " is square" << endl;

}

else cout << K << " isn't square" << endl;

return isSquare;

}

int main()

{

float num;

cout << "Enter number please" << endl;

cin >> num;

isSquare(num);

return 0;

}

**Завдання 3**

Описати функцію аddLeftDigit(D, K), додаючи до запису цілого додатного числа K зліва цифру D (D – вхідний параметр цілого типу, що лежить у діапазоні 1–9, K – параметр цілого типу, що є одночасно вхідним та вихідним). За допомогою цієї функції послідовно додайте до заданого числа K цифри D1 і D2, виводячи результат кожного додавання на екран.

1. **Математична постановка задачі (МПЗ).**

*Вхідні дані*: K, D –змінні цілого типу.

*Вихідні дані*: K - результат цілого типу.

*Математична модель задачі*: temp =10; поки К більше temp: temp \*= 10; K += D \* temp;

Подання МПЗ у вигляді таблиці:

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| K, D –змінні цілого типу. | temp =10; поки К більше temp: temp \*= 10; K += D \* temp; | K - результат цілого типу. |

Якщо подати функцію як “чорну скриню”, маємо:



Рисунок 7 – Чорна скриня функції задачі 3.

Якщо подати головну програму як “чорну скриню”, маємо:



Рисунок 8 – Чорна скриня головної функції задачі 3.

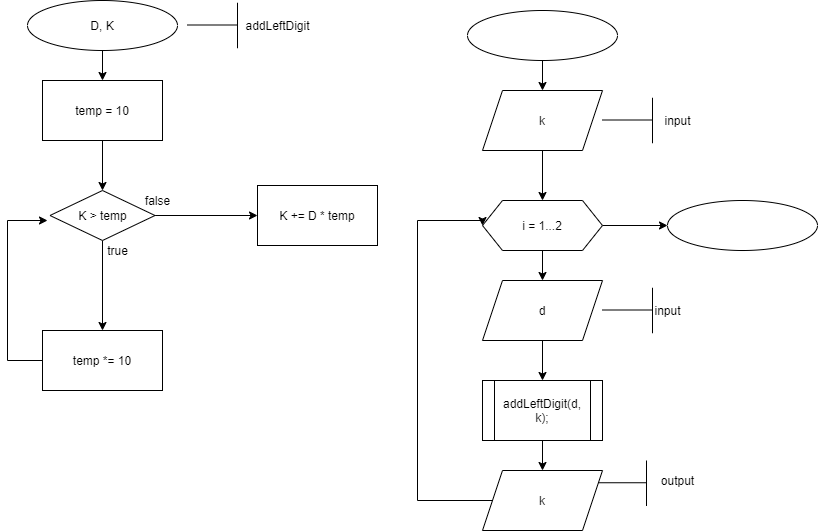
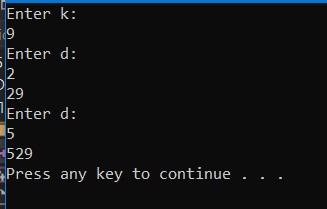


Рисунок 9 - Схема алгоритму задачі 3.

1. **Тестові приклади.**

Вхідні дані: k = 9, d1 = 2, d2 = 5;

Вихідні дані: 29, 529;



1. **Текст програми:**

#include <iostream>

using namespace std;

void addLeftDigit(int D, int\* K) { //змінні покажчики

int temp = 10;

while (\*K > temp) temp \*= 10;

\*K += D \* temp;

}

int main()

{

int k;

cout << "Enter k: " << endl;

cin >> k;

for (int i = 1; i <= 2; i++) {

int d;

cout << "Enter d: " << endl;

cin >> d;

addLeftDigit(d, &k);

cout << k << endl;

}

return 0;

}

**Завдання 4**

Описати функцію shiftLeft3(A, B, C), яка виконує лівий циклічний зсув: значення A переходить в B, значення B – в C, значення C – в A (A, B, C – дійсні параметри, що є одночасно вхідними та вихідними). За допомогою цієї функції виконайте лівий циклічний зсув для двох наборів з трьох чисел: (A1, B1, C1) та (A2, B2, C2).

1. **Математична постановка задачі (МПЗ).**

*Вхідні дані*: A, B, C –змінні дійсного типу.

*Вихідні дані*: A, B, C - результат дійсного типу.

*Математична модель задачі*: temp = a; a = c; c = b; b = temp;

Подання МПЗ у вигляді таблиці:

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| A, B, C –змінні дійсного типу. | temp = a; a = c; c = b; b = temp; | A, B, C - результат дійсного типу. |

Якщо подати функцію як “чорну скриню”, маємо:



Рисунок 10 – Чорна скриня функції задачі 4.

Якщо подати головну програму як “чорну скриню”, маємо:



Рисунок 11 – Чорна скриня головної функції задачі 4.

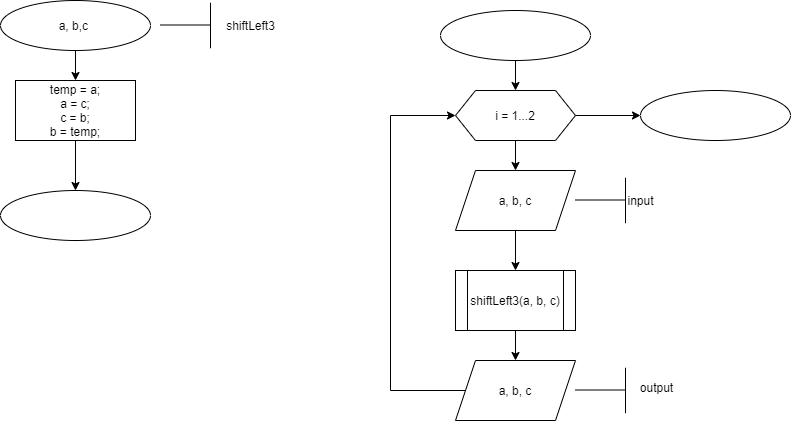
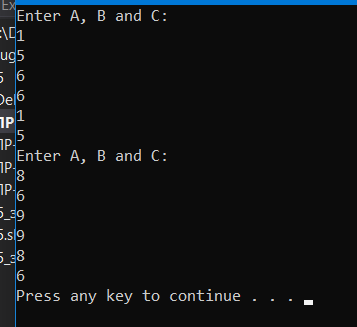


Рисунок 12 - Схема алгоритму задачі 4.

1. **Тестові приклади.**

Вхідні дані: A= 1, B1 = 5, C1 = 6 ; A2= 8, B2 = 6, C2 = 9 ;

Вихідні дані: A1= 6, B1 = 1, C1 = 5 ; A2= 9, B2 = 8, C2 = 6;



1. **Текст програми:**

#include <iostream>

using namespace std;

void shiftLeft3(float& a, float& b, float& c) {

float temp = a;

a = c;

c = b;

b = temp;

}

int main()

{

for (int i = 1; i <= 2; i++)

{

float a, b, c;

cout << "Enter A, B and C: " << endl;

cin >> a >> b >> c;

shiftLeft3(a, b, c);

cout << a << endl << b << endl << c << endl;

}

return 0;

}

**Завдання 5**

Описати перевантажені функції та функцію-шаблон (див. завдання 3 до лабораторної роботи) для знаходження суми квадратів двох заданих чисел.

1. **Математична постановка задачі (МПЗ).**

*Вхідні дані*: a, b –змінні будь-якого типу(використовується шаблон функції) .

*Вихідні дані*: sum - результат будь-якого типу(в залежності від типу вхідних даних).

*Математична модель задачі*: a\*a + b\*b

Подання МПЗ у вигляді таблиці:

|  |  |  |
| --- | --- | --- |
| Вхідні дані | Дії | Вихідні дані |
| a, b –змінні будь-якого типу(використовується шаблон функції) . | a\*a + b\*b | sum - результат будь-якого типу(в залежності від типу вхідних даних). |

Якщо подати функцію як “чорну скриню”, маємо:



Рисунок 13 – Чорна скриня функції задачі 5.

Якщо подати головну програму як “чорну скриню”, маємо:



Рисунок 14 – Чорна скриня головної функції задачі 5.

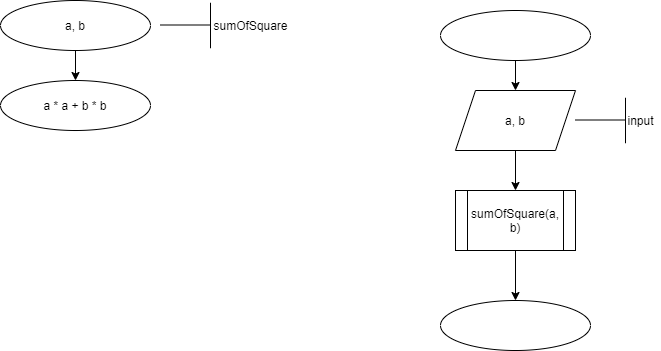
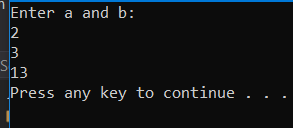


Рисунок 15 - Схема алгоритму задачі 5.

1. **Тестові приклади.**

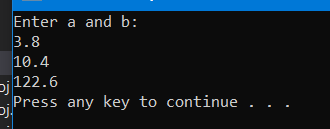
Вхідні дані: a = 2, b = 3;

Вихідні дані: 13;



Вхідні дані: a = 3.8, b = 10.4;

Вихідні дані: 122.6;



1. **Текст програми:**

#include <iostream>

using namespace std;

// приклади перевантаження

//int sumOfSquare(int a, int b) {//перевантаження для цілочисельного типу

// return a \* a + b \* b;

//}

//

//long sumOfSquare(long a, long b) {//перевантаження для типу long

// return a \* a + b \* b;

//}

//

//float sumOfSquare(float a, float b = 5.5) {//перевантаження для типу float + параметри за замовчуванням

// return a \* a + b \* b;

//}

// шаблон функції

template <typename T>

const T& sumOfSquare(const T& a, const T& b)

{

return a \* a + b \* b;

}

int main()

{

int a = 2, b = 3;

float a1 = 3.8, b2 = 10.4;

cout << sumOfSquare(a, b) << endl;

cout << sumOfSquare(a1, b2) << endl;

return 0;

}