

# Лабораторна робота №7 Функції

## 1 Вимоги

### 1.1 Розробник

-Носов Микола  
-студент групи кіт 1206  
-27.11.2020

### 1.2 Мета:

Навчитися працювати із функціями

### 1.3 Індивідуальне завдання:

**На оцінку “відмінно”.** Зокрема загального завдання необхідно виконати наступне:

- реалізувати функцію, що визначає, скільки серед заданої послідовності чисел таких пар, у котрих перше число менше наступного, використовуючи функцію з варіативною кількістю аргументів.

## 2 Опис програми

### 2.1 Функціональне призначення

Програма призначена для:

- знаходження НОД 2 чисел;
- перевірки на білета на “щастливість”;
- перевірки числа, і визначення, чи є воно простим;
- перевірки числа, і визначення, чи є ідеальним;
- знаххоження квадратного кореня числа;
- центрування слів у строчці;

### 2.2 Логічна структура проекту

Структура проекту (див рис. 1)



Рисунок 1 - робота утиліти tree

## 1. Функція знаходження НОД (див рис. 2)

```
5 int nod1( int a, int b )  
6 {  
7     int nod;
```

Рисунок 2 - код функції

### Призначення :

-Знаходження найбільшого спільного дільника ;

### Аргументи :

-2 числа;

### Опис роботи :

-визначення, яке число число є більшим;

-ділення більшого числа на менше по %, доки вони  $\neq 0$ ;

-сумуємо 2 числа;

## 2. Функція визначення щасливого білетика (див рис. 3)

```
34 int lucky_ticket(int n)  
35 {  
36     int x2 = 0;  
37     int x3 = 0;  
38     int j = 0;
```

Рисунок 3 - код функції

### Призначення :

-Визначення, чи є білетик щасливим;

### Аргументи :

-6-ти значне число;

### Опис роботи :

-розбиття 6-ти значного числа на 6 цифр;

-сума перших трьох чисел;

-сума чисел, що залишилися;

-якщо суми однокові - білетик щасливий;

### 3. Функція визначення простого числа (див рис. 4)

```
74 int prime_digit(int a12)
75 {
76     int result;
```

*Рисунок 4 - код функції*

#### **Призначення :**

-Визначення, чи є число простим;

#### **Аргументи :**

-Число

#### **Опис роботи :**

-Ділити число на всі числа від 1 до квадратного кореня заданого числа;

-Якщо число ділиться на ціло - число не просте, в іншому випадку -

просте;

### 4. Функція визначення ідеально числа (див рис.5)

```
int ideal_digit(int bi )
{
    int pi = 1;
    int ci = 0;
```

*Рисунок 5 - код функції*

#### **Призначення :**

-Визначення, чи є число ідеальним;

#### **Аргументи :**

-Число;

#### **Опис роботи :**

-знаходження дільників числа;

-якщо сума дільників = самому числу - воно ідеальне;

## 5. Функція знаходження квадратного кореня (див рис 6)

```
float sqrt1(float chislo1)
{
    int chislo;
    float sqrt;
    float podkor_proiz;
    float chel_proiz;
    char s = '√';
    float digit = chislo1;
```

Рисунок 6 - код функції

### Призначення :

-Знаходження квадратного кореня;

### Аргументи :

-Число

### Опис роботи :

-знаходження дільників числа;

-попарне винесення дільників числа із під знаку кореня;

-знаходження кореня за теоремою дотичних Ньютона;

Отримаємо відповідь у такому вигляді (див рис 7 ):

$10\sqrt{2} \sim 14.142136$

Рисунок 7 - квадратний корінь з числа 200

## 6. Функція центрування слова у рядку (див рис 8)

```
void centrovka(char mass[])
{
    int centr = 0;
    int shetBukv = 0;
    int firstChar;
```

Рисунок 8 - функція, що нічого не повертає

### Призначення :

-Центрування слова або словосполучення у рядку;

### Аргументи :

-Слово, що потребує цетрування;

### Опис роботи :

-Знаходження першої букви;

- Підрахунок кількості букв;
  - Розрахунок необхідної кількості символів центрування;
  - Заповнення рядка буквами та символами “\_”;
- Отримаємо відповідь у такій формі (див рис 9)

`__Ivan Borov__`

Рисунок 9 - результат роботи функції 6

7 Функція з варіативною кількістю аргументів (див рис 10)

```
int couples(int var, ...)
{
    int count = 0;
    int value = 0;
    va_list vl;
    va_start(vl, var);
    int mass[var];
```

Рисунок 10 - функція

#### **Призначення :**

- Знаходження кількості пар чисел, перше число якої менше за наступне;

#### **Аргументи :**

- Ряд чисел;

#### **Опис роботи :**

- Програма порівнює 2 числа, що стоять поруч;
- Якщо число менше наступного - образується пара;

### **3 Варіанти використання**

Для обчислення результатів ми використаємо gdb - дебагер, влаштований в утиліту gcc.

Щоб подивитися на результат, ми повинні:

- при компіляції вказати рівень інформації для відлатки за допомогою -g;
- відкомпілювати наш файл з кодом;
- запустити його бінарний файл у відлагоднику;
- вказати ім'я функції, як точку зупину щоб подивитись на значення змінних;
- запускаємо програму та вводимо: “info locals”;

**Висновок:**

Протягом виконання даної лабораторної роботи ми навчилися розбивати нашу програму на підпрограми за допомогою функцій, а також познайомились з функцією, яка має варіативну кількість аргументів.