**Лабораторна робота №8 Блок схеми аглоритмів**

1 Вимоги

1.1**Розробник**

-Носов Микола

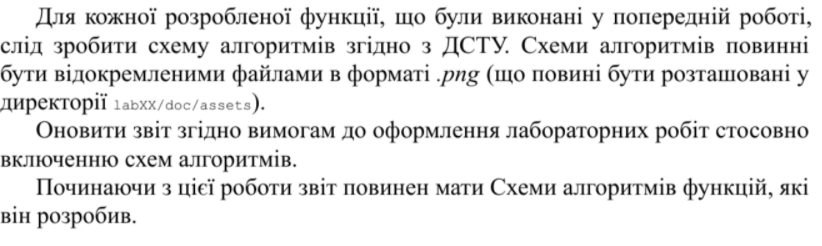
-студент групи кіт 120б

-8.12.2020

**1.2 Мета:**

Навчитися складати схеми алгоритмів

1.3 **Загальне завдання**



2 Опис програми

**2.1 Функціональне призначення**

Програма призначена для**:**

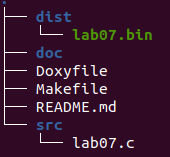
-перевірки на білета на “щастливість”;

-центрування слів у строчці;

**-**підрахунку кількості пар, де пешре число меньше наступного у вхідній послідовності;

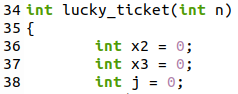
**2.2 Логічна структура проекту**

Структура проекту (див рис. 1)



*Рисунок 1* - *робота утиліти tree*

1. Функція визначення щасливого білетика (див рис. 2)



*Рисунок 2 - фрагмент коду*

**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;

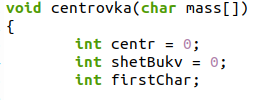
-якщо суми однокові - білетик щасливий;

**Схема алгоритму** (див рис 3) :



*Рисунок 3 - креслення алгоритму*

1. Функція центрування слова у рядку (див рис 4)



*Рисунок 4 - 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 кількости символів центрування;

-Заповнення рядка буквами та символами “\_”;

Отримаємо відповідь у такій формі (див рис 5)



*Рисунок 5 - результат роботи функції 6*

**Схем алгоритму** (див рис 6, 7, 8):



*Рисунок 6 - 1 частина креслення алгоритму*

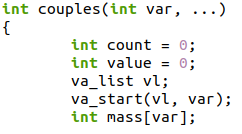
**

*Рисунок 7 - 2 креслення алгоритму*

**

*Рисунок 8 - 3 частина креслення алгоритму*

3. Функція з варіативною кількістю агрументів (див рис 9)



*Рисунок 9 - фрагмент коду*

**Призначення :**

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

**Аргументи :**

**-**Ряд чисел;

**Опис роботи :**

*-*Программа порівнює 2 числа, що стоять поруч;

*-*Якщо число меньше наступного - образовується пара;

**Схема алгоритму** (див рис 10):



*Рисунок 10 - креслення алгоритму*

**3 Варіанти використання**

Для обчислення результатів мы використовємо gdb - дебагер, влаштований в

утиліту gcc.

Щоб подивитися на резутат, ми повинні:

-при компіляції вказати рівень інформації для відлатки за допомогою -g; -відкомпілювати наш файл з кодом;

-запустити його бінарний файл у відлагоднику;

-вказати им’я функції, як точку зупину щоб подивитьсь на значення змінних;

-запускаємо програму та вводимо: “info locals”;

**Висновок:**

Протягом виконання даної лабораторної роботи ми навчилися зображати роботу нашого алгоритму у вигляді схеми.