**Лабораторна робота №11 Вступ до показчиків**

1 Вимоги

1.1**Розробник**

-Носов Микола

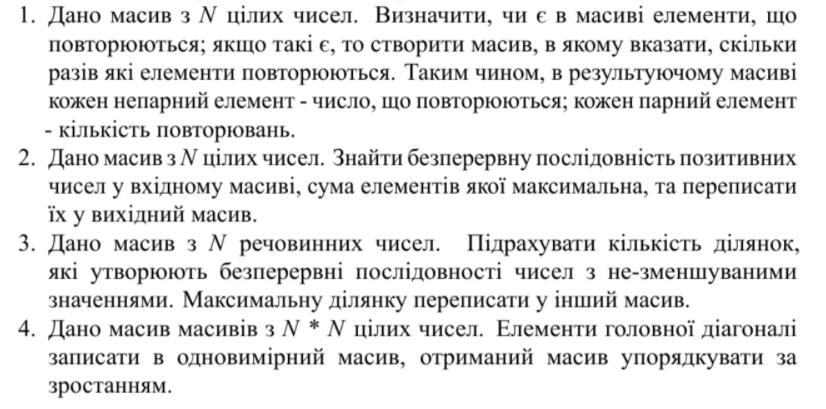
-студент групи кіт 120б

-23.12.2020

**1.2 Мета:**

Навчитися працювати з показчиками, виділенням пам’яті

1.4 **Індивідуальне завдання:**



2 Опис програми

**2.1 Функціональне призначення**

Програма призначена для**:**

-знаходження числа повторювань елементів массиву;

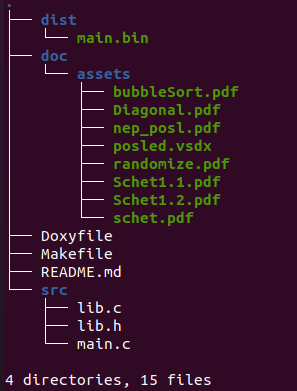
-знаходження безперервної додатньої послідновності, сумма елементів якої найбільша ;

-знаходження максимальної ділянки чисел, с незменшуванними значеннями;

-упорядкування за зростанням елементів головної діагоналі матриці;

**2.2 Логічна структура проекту**

Структура проекту (див рис. 1)



*Рисунок 1* - *робота утиліти tree*

1. **Задача 1** - знахождення числа повторювань елементів массиву

**Призначення :**

-знайти число повторювань елементів массиву

**Агрументи :**

**-** массив чисел

**Опис роботи:**

**-** массив сортуеться за збільшенням

- доки перший елемент дорінює наступному, числа повторюються

**Схеми алгоритмів**(див рис 2, 3, 4)**:**

****

*Рисунок 2 - бульбашкове сортування*

**

*Рисунок 3 - заповлення массива випадковими числами*

**

*Рисунок 4 - основний алгоритм задачі*

**Задача 2**

**Призначення :**

- знаходження безперервної додатньої послідновності, сумма елементів якої найбільша

**Агрументи :**

**-** числова послідовність

**Опис роботи:**

- визначення послідовностей

- заповнення 2 - виморного массиву елементами всіх послідвностей

- підрахунок сумми всіх послідовностей та порівняння

**Блок схеми задачі** :



*Рисунок 5 - основний алгоритм задачі*



*Рисунок 6 - підрахунок кількості послідовностей*



*Рисунок 7 - продовження основного циклу задачі 2*

**Задача 4**

**Призначення :**

- упорядкування за зростанням елементів головної діагоналі матриці

**Агрументи :**

**-** 2 - вимірний массив

**Опис роботи:**

- елементи головної діагоналі переписуються у звичанйи массив

- елементи сортуються за зростанням

**Схема алгоритму** (див рис 8 , 2) **:**

****

*Рисунок 8 - основний алгоритм задачі*

**2.3** Важливі фрагменти програми

int \* block = calloc(b, sizeof(int)); - виділення пам’яті

void schet (int block[], int mass[], int \*chetmas); - передача змінної та массиву в функцію

**3** Варіанти використання

Для обчислення результатів мы використовємо gdb - дебагер, влаштований в

утиліту gcc.

Щоб подивитися на резутат, ми повинні:

- зібрати проект за домогою make

- 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** :

В умовах даної лабораторної роботи ми навчилися працювати с динамічним відленням памьяті та вказівниками.