**ЗВІТ**

**до лабораторної роботи № < 12.6 >**

**« Черги »**

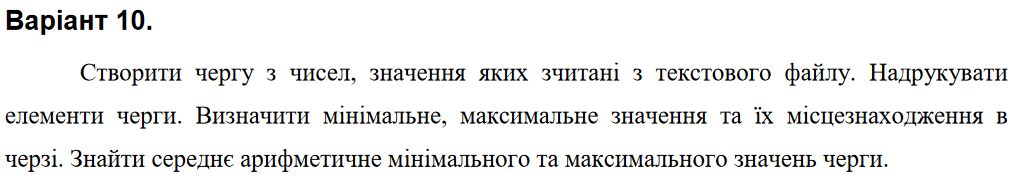
**з дисципліни**

**«Алгоритмізація та програмування»**

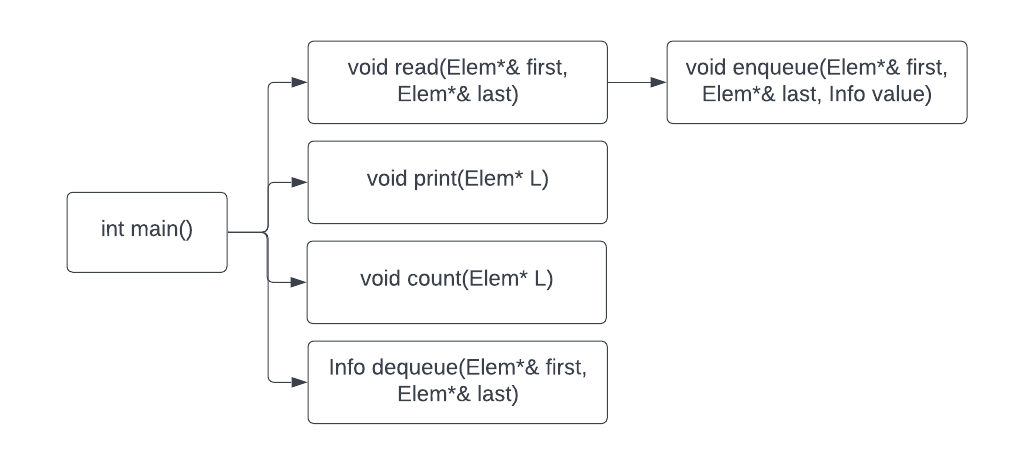
**студента групи ІН-105Б**

**Горанова Анастаса**

**Умова завдання:**



**Структурна схема:**



**Текст програми:**

// 12.6.cpp

// < Горанов Анастас >

// Лабораторна робота № 12.6

// Черги

// Варіант 10

#include <iostream>

#include <fstream>

#include <string>

using namespace std;

typedef int Info;

struct Elem

{

Elem\* link;

int info;

};

void enqueue(Elem\*& first, Elem\*& last, Info value);

Info dequeue(Elem\*& first, Elem\*& last);

void read(Elem\*& first, Elem\*& last);

void print(Elem\* L);

void count(Elem\* L);

int main()

{

Elem\* first = NULL,

\* last = NULL;

read(first, last);

cout << endl;

print(first); // вивід черги

cout << endl;

count(first);

cout << endl;

cout << "Queue: " << endl << endl;

while (first != NULL)

cout << dequeue(first, last) << " ";

cout << endl;

return 0;

}

void enqueue(Elem\*& first, Elem\*& last, Info value)

{

Elem\* tmp = new Elem;

tmp->info = value;

tmp->link = NULL;

if (last != NULL)

last->link = tmp;

last = tmp;

if (first == NULL)

first = tmp;

}

Info dequeue(Elem\*& first, Elem\*& last)

{

Elem\* tmp = first->link;

Info value = first->info;

delete first;

first = tmp;

if (first == NULL)

last = NULL;

return value;

}

void read(Elem\*& first, Elem\*& last)

{

ifstream f("ap12.txt");

Info symbol;

while (!f.eof()) {

f >> symbol;

enqueue(first, last, symbol);

}

}

void print(Elem\* L)

{

while (L != NULL)

{

cout << L->info << " ";

L = L->link;

}

cout << endl;

}

void count(Elem\* L)

{

Info min = L->info;

Info max = L->info;

Info min\_pos = 0;

Info max\_pos = 0;

Info k = 0;

while (L->link != NULL)

{

k++;

if (L->info < min) {

min = L->info;

min\_pos = k;

}

else if (L->info > max) {

max = L->info;

max\_pos = k;

}

L = L->link;

}

if (L != NULL) {

k++;

if (L->info < min) {

min = L->info;

min\_pos = k;

}

else if (L->info > max) {

max = L->info;

max\_pos = k;

}

}

cout << "Minimum " << min << " be in position " << min\_pos << endl;

cout << "Maximum " << max << " be in position " << max\_pos << endl;

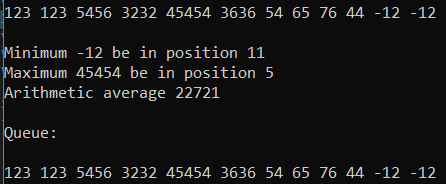
cout << "Arithmetic average " << (max + min) / 2 << endl;

}

**Посилання на git-репозиторій з проектом:**

*https://github.com/StassNG/12.6*

**Результати програми та unit-тесту:**

****

**Висновок:** я навчився опрацьовувати черги.