**ЗВІТ**

**до лабораторної роботи № < 12.2 >**

**« Опрацювання лінійного однонаправленого списку »**

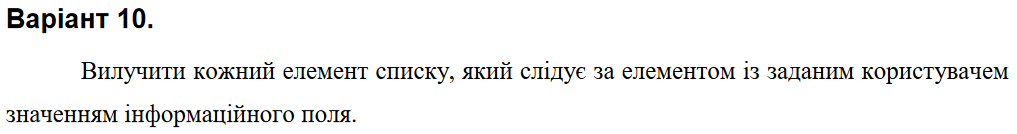
**з дисципліни**

**«Алгоритмізація та програмування»**

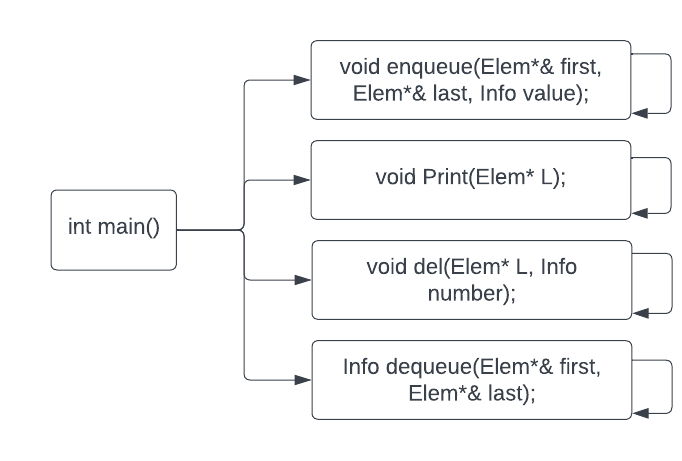
**студента групи ІН-105Б**

**Горанова Анастаса**

**Умова завдання:**



**Структурна схема:**



**Текст програми:**

// 12.2IT.cpp

// < Горанов Анастас >

// Лабораторна робота № 12.2IT

// Опрацювання лінійного однонаправленого списку

// Варіант 10

#include <iostream>

using namespace std;

typedef int Info; // Cтворення псевдоніму типу

struct Elem

{

Elem\* link;

Info info;

};

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

void Print(Elem\* L);

void del(Elem\* L, Info number);

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

int main()

{

Elem\* first = NULL,

\* last = NULL;

for (int i = 1; i <= 10; i++)

enqueue(first, last, i);

Print(first); // Вивід

cout << endl;

Info number;

cout << "Enter a number after which all items will be deleted: "; cin >> number;

del(first, number); // Видалення компонентів

while (last != NULL) {

cout << dequeue(first, last) << " "; // Вивід та видалення черги

}

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 Print(Elem\* L)

{

while (L != NULL)

{

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

L = L->link;

}

cout << endl;

}

void del(Elem\* L, Info number)

{

while (L != NULL && L->link != NULL) {

if (L->link->info > number) {

Elem\* tmp = L->link->link;

delete L->link;

L->link = tmp;

}

else

L = L->link;

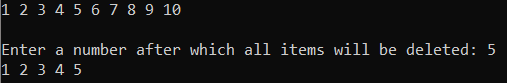
}

}

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

*https://github.com/StassNG/12.2IT*

**Результати програми та unit-тесту:**

****

**Висновок:** Я навчився опрацьовувати лінійні однонаправлені списки.