

**Лабораториска вежба 4 – Единечно поврзани листи**

Име и презиме	Број на индекс	Група

**Задача 1:**

Да се напише функција за работа со единечно поврзани кружни листи, која на влез добива две пополнети листи. Функцијата треба да ги измине истовремено листите и доколку наиде на јазли со иста вредност, јазолот од втората листа да се префрли во првата листа (во втората листа повеќе да го нема) и тоа пред моментално разгледуваниот јазол во првата листа. Изминувањето се прави се додека има јазли и во двете листи. Изминувањето на листите да се направи во глобална функција *promeni*.

Како дел од програмата да се напишат структури за имплементација на јазол и листа и функциите за поддршка на листата: функција за иницијализација, функции додавање и бришење на елемент на почеток од листата, додавање и бришење на елемент на крај од листата, бришење на цела листа, печатење на елементите од листата и бришење на јазол (адресата на јазолот кој треба да се брише се добива на влез од функцијата).

Влез:

Lista1: |10|->|2|->|5|->|7|->|13|

Lista2: |5|->|2|->|7|->|7|

Излез:

Lista1: |10|->|2|->|2|->|5|->|7|->|7|->|13|

Lista2: |5|->|7|

Објаснување: при изминувањето први елементи кои се исти се јазлите со вредности |2| и |2| па, јазолот со вредност |2| од втората листа се префрла во првата листа така што да претходи на јазолот со вредност |2|. Сега следни при изминувањето ќе бидат јазлите со вредности |5| и |7|, за првата листа и за втората листа соодветно. Префрлањето може да се реализира и со креирање и бришење на јазол или со промена на линкови во постојните јазли (двете решенија се прифатливи).

Забелешка: Почетниот код од задачата е даден во продолжение:

```
#include <iostream>
#include <stdlib.h>
#include <string>
```

```
using namespace std;
```

```
int main(){
    KLista L1, L2;
```

```
L1.init();
L2.init();

L1.dodadiPrv(10);
L1.dodadiPosleden(2);
L1.dodadiPosleden(5);
L1.dodadiPosleden(7);
L1.dodadiPosleden(13);

L2.dodadiPosleden(7);
L2.dodadiPrv(7);
L2.dodadiPrv(2);
L2.dodadiPrv(5);

promeni(L1, L2);

L1.pechati();
L2.pechati();

L1.brishiLista();
L2.brishiLista();

return 0;
}
```