

Blok 3 - Jednostruko povezane i kružne liste

Napomena: Pri rešavanju sledećih zadataka, iskoristiti biblioteku za rad sa listama koja je radjena na prethodnim časovima. Biblioteka se nalazi u nastavnim materijalima. Gde god je moguće preferirajte rekurzivni način rešavanja.

Zadatak 1 (5 poena) Napisati funkciju koja određuje dužinu unešene liste: `int duzina(EL* plist)`. Test primer:

```
6 2 4 7 2 6 19 3 5 -1
9
```

Zadatak 2 (10 poena) Napisati funkciju koja pravi listu od cifara zadatog broja: `EL* lista_od_broja(int n)`. Test primer:

```
4232323
pravi listu: 4->2->3->2->3->2->3->NULL
i ispisuje je: 4 2 3 2 3 2 3
```

Zadatak 3 (10 poena) Spojiti dve sortirane liste unete sa ulaza u treću listu koja je takođe sortirana: `EL* spoji_sortirano(EL* plist1, EL* plist2)`. Dozvoljen je samo jedan prolaz kroz unete liste (Napomena: zadatak je sličan onom sa časa u kojem su se liste spajale uplitanjem). Test primer:

```
1 3 4 7 19 78 198 -1
1 2 2 5 6 7 18 29 200 -1
1 2 2 3 4 5 6 7 18 19 29 78 198 200
```

Zadatak 4 (10 poena) Napisati rekurzivnu funkciju koja eliminiše duplikate iz zadate liste: `EL* eliminisi_duplikate(EL* plist)`. Test primer:

```
2 46 6 3 6 2 6 8 2 1 3 5 8 -1
2 46 6 3 8 1 5
```

Zadatak 5 (10 poena) Na ulazu su dve liste. Napraviti novu treću listu takvu da sadrži zajedničke elemente date dve liste `EL* presek(EL* plist1, EL* plist2)`. (Napomena: praviti nove elemente za treću listu, ne prespajati postojeće). Test primer:

```
5 6 2 3 6 4 8 5 27 -1
7 3 4 2 7 9 4 2 7 -1
3 4 2
```

Zadatak 6 (15 poena) Učitati listu, napraviti od nje kružnu, i potom napraviti funkciju: `void ubaci_zbir_prethodnih_k(EL* plist, int k)`, koja ciklično prolazi kroz k elemenata i potom na aktivnoj poziciji umeće element koji ima vrednost zbira tih k elemenata. Program se zaustavlja nakon što upiše 10 vrednosti na ovaj način. Test primer:

```
5 6 3 -1
2 //k
5 6 11 3 //posle prvog upisa (5+6=11)
5 8 6 11 3 //drugi upis (3+5=8)
5 8 6 11 17 3 //treci upis (6+11=17)
...
```