

Vježba 3

Cilj vježbe 3 je upoznavanje sa strukturom podataka stek, pri čemu će studenti za pripremu uraditi implementaciju stek strukture podataka, a zatim na tutorijalu primijeniti implementaciju za rješavanje nekoliko problema

Zadatak 1.

U što manje linija koda napraviti funkciju:

```
bool provjeri_zagrade(string tekst)
```

koja provjerava da li je svaka otvorena zagrada u primljenom stringu korektno zatvorena. Dozvoljena su tri tipa zagrada: obične (), uglaste [] i vitičaste {}. Funkcija treba vratiti true za string kao što je: “[()]{ }{(OO)O}”. Međutim, treba vratiti false ako se jedna vrsta zagrada ukrštava sa drugim, npr. ako string glasi “[()]” treba vratiti false. U stringu se mogu javljati i drugi karakteri (koje treba zanemariti).

Pri provjeri nije potrebno pridržavati se pravila iz matematike da se uglasta nalazi unutar vitičaste a obična unutar uglaste, samo se provjerava da li je svaki tip otvorene zagrade korektno zatvoren i da li ima ukrštanja.

Zadatak 2.

Napisati funkciju:

```
int presjek(Stek<Tip> s1, Stek<Tip> s2)
```

koja prima dva sortirana generička steka (tako da se na vrhu steka nalazi najmanji element), a vraća broj elemenata u njihovom presjeku odnosno broj elemenata koji se javljaju u oba steka. Pretpostaviti da se elementi ne ponavljaju u istom steku.

Primjer: Neka su u prvom steku elementi A C D F, a u drugom B C D E F G. Funkcija treba vratiti broj 3 pošto postoje tri elementa koji se nalaze u oba steka (a to su C D F).

Zadatak 3.

Napisati kod funkcije čiji je prototip

```
bool palindrom(const Lista<char>& rijec)
```

koja vraća **true** ako riječ sadržana u Listi tipa char predstavlja palindrom (jednaka je s početka i kraja), a **false** ako ne predstavlja. Primjer palindroma:

RATAR

ANAVOLIMILOVANA

Pretpostaviti da string sadrži isključivo velika slova engleske abecede.

Napomena: Za ovaj zadatak je potrebno koristiti klase Lista razvijene na prošlom tutorijalu. Ako ima nepopravljenih grešaka u ovim klasama, potrebno ih je najprije popraviti.

Zadatak 4.

Napisati funkciju:

`int spoji(Stek<int> s1, Stek<int> s2, Stek<int>& s3)`

Funkcija prima stek s1 koji je sortiran rastući, stek s2 koji je sortiran opadajući i s3 koji je prazan.

Potrebno je prebaciti elemente iz steka s1 i s2 u stek s3 tako da stek bude sortiran rastući pri čemu je dozvoljeno koristiti samo tri data steka, tj. nisu dozvoljene dodatne kolekcije.

Primjer: Neka je stek s1: {1, 3, 5, 8} a s2: {9, 6, 4, 2} (prvi element je na dnu steka). Nakon pozvane funkcije spoji stek s3 treba da bude {1, 2, 3, 4, 5, 6, 8, 9}.