

Prvi zadatak (20 bodova)**1. grupa**

Napišite parametriziranu strukturu **stablo** koja reprezentira stablo čiji čvorovi osim podatka imaju i identifikator (tipa **int**). Svaki čvor može imati nula ili više djece (čvorovi ne moraju imati jednak broj djece). Svaki čvor sadrži polje pokazivača na svoju djecu. Polja moraju biti dinamički alocirana, pri tome pripazite da se pri svakom dodavanju/izbacivanju čvora ne mora ponovno realocirati memorija. Dodajte sve potrebne varijable u strukturu da bi to mogli ostvariti. Struktura će se koristiti sa standardnim C++ tipovima nad kojima je definiran operator $<$.

Za navedenu strukturu implementirajte:

- konstruktor bez parametara
- konstruktor koji kao parametre prima element i njegov identifikator te stvara stablo koje se sastoji samo od korijena koji sadrži zadane podatke.
- Konstruktor koji kao ulazni parametar prima **string**. Konstruktor stvara stablo koje se sastoji od korijena, njegove djece i djece njegove djece. String na početku sadrži niz brojeva razmaknut razmakom, gdje prvi broj definira broj djece korijena a svaki sljedeći broj djece jednog djeteta od korijena. U nastavku se nalaze elementi stabla odvojeni znakom ';'. Djeca svakog čvora moraju biti sortirana uzlazno te im se slijedno dodjeljuje identifikator.

Primjer: Za ulazni string "2 3 1;4;1;12;5;6;9;555" treba stvoriti stablo:

(4 , 0)

(1 , 1) (12 , 2)

(5 , 3) (6 , 4) (9 , 5) (555 , 6)

Uputa: provjeru tipa parametra T unutar konstruktora možete raditi korištenjem:

```
#include <type_traits>
```

```
std::is_same<T, int>::value //primjer za int
```

- Napišite funkciju **int ubaci(int iden, T pod, int mid);** koja ubacuje element *pod* u stablo kao dijete čvoru s identifikatorom *iden*. Varijabla *mid* sadrži vrijednost najvećeg identifikatora čvora stabla. Funkcija vraća 1 ako je čvor uspješno ubačen, inače vraća 0 (ne postoji čvor s identifikatorom *iden*).
- Napišite funkciju **void maxId(int *m);** koja sprema vrijednost najvećeg identifikatora čvora stabla u varijabilni argument *m*.
- Napišite funkciju **int izbaci(int iden);** koja izbacuje čvor s identifikatorom *iden*, njegovu djecu i svu djecu njegove djece iz stabla (cijelo pod stablo). Funkcija vraća 1 ako je izbacivanje uspješno provedeno, inače vraća 0 (čvor nije bio element stabla).
- Napišite funkciju **void ispisi_stablo(std::queue<stablo<T>*> q);** koja ispisuje stablo razinu po razinu, čvorovi iste razine su odmaknuti razmakom, svaka razina je zapisana u svom retku (vidi primjer stabla gore).

Napomena: zabranjeno je korištenje STL-a za implementaciju svih točaka osim zadnje, u njoj smijete koristiti red kao ulazni parametar i još jedan takav red kao pomoćnu varijablu.

PREZIME I IME:

MATIČNI BROJ STUDENTA:

BODOVI:

Prvi zadatak (20 bodova)**2. grupa**

Napišite parametriziranu strukturu **stablo** koja reprezentira stablo čiji čvorovi osim podatka imaju i identifikator (tipa **int**). Svaki čvor može imati nula ili više djece (čvorovi ne moraju imati jednak broj djece). Svaki čvor sadrži polje pokazivača na svoju djecu. Polja moraju biti dinamički alocirana, pri tome pripazite da se pri svakom dodavanju/izbacivanju čvora ne mora ponovno realocirati memorija. Dodajte sve potrebne varijable u strukturu da bi to mogli ostvariti. Struktura će se koristiti sa standardnim C++ tipovima nad kojima je definiran operator `<`.

Za navedenu strukturu implementirajte:

- konstruktor bez parametara
- konstruktor koji kao parametre prima element, početni identifikator i broj elemenata. Konstruktor stvara stablo koje kao korijen ima element (*el*) s početnim identifikatorom (*pi*), a kao djecu čvorove s vrijednosti *el*, *el+el*, *el+el+el* itd. i identifikatorima *pi+1*, *pi+2*, *pi+3* itd.
- Konstruktor koji kao ulazni parametar prima **string**. Konstruktor stvara stablo koje se sastoji od korijena, njegove djece i djece njegove djece. String na početku sadrži niz brojeva razmaknut razmakom, gdje prvi broj definira broj djece korijena a svaki sljedeći broj djece jednog djeteta od korijena. U nastavku se nalaze elementi stabla odvojeni znakom ';'. Djeca svakog čvora moraju biti sortirana silazno te im se slijedno dodjeljuje identifikator.

Primjer: Za ulazni string "2 3 1;4;1;12;5;6;9;555" treba stvoriti stablo:

(4 , 0)

(12 , 1) (1 , 2)

(9 , 3) (6 , 4) (5 , 5) (555 , 6)

Uputa: provjeru tipa parametra T unutar konstruktora možete raditi korištenjem:

```
#include <type_traits>
```

```
std::is_same<T, int>::value //primjer za int
```

- Napišite funkciju **int ubaci(int iden, T pod, int mid);** koja ubacuje element *pod* u stablo kao dijete čvoru s identifikatorom *iden*. Varijabla *mid* sadrži vrijednost najvećeg identifikatora čvora stabla. Funkcija vraća 1 ako je čvor uspješno ubačen, inače vraća 0 (ne postoji čvor s identifikatorom *iden*).
- Napišite funkciju **void maxId(int *m);** koja sprema vrijednost najvećeg identifikatora čvora stabla u varijabilni argument *m*.
- Napišite funkciju **int izbaci(int iden);** koja izbacuje čvor s identifikatorom *iden*, njegovu djecu i svu djecu njegove djece iz stabla (cijelo pod stablo). Funkcija vraća 1 ako je izbacivanje uspješno provedeno, inače vraća 0 (čvor nije bio element stabla).
- Napišite funkciju **void ispisi_stablo(std::queue<stablo<T>*> q);** koja ispisuje stablo razinu po razinu, čvorovi iste razine su odmaknuti razmakom, svaka razina je zapisana u svom retku (vidi primjer stabla gore).

Napomena: zabranjeno je korištenje STL-a za implementaciju svih točaka osim zadnje, u njoj smijete koristiti red kao ulazni parametar i još jedan takav red kao pomoćnu varijablu.

PREZIME I IME:

MATIČNI BROJ STUDENTA:

BODOVI:

Drugi zadatak (15 bodova)**1. grupa**

Sezona utrka formule 1 sadrži k utrka. Svaka utrka se pamti kao vektor trojki koje sadrže ime vozača, ime tima i vrijeme vožnje u minutama. Pretpostavljamo da su sva vremena u utrci različita. Trojku uredite po želji, međutim koristite konzistentno u rješavanju zadatka. Ukoliko sa ulaza učitavamo informacije o utrkama na sljedeći način:

```
vozač1,1;tim1,1;vrijeme1,1;vozač2,1;tim2,1;vrijeme2,1;...;vozačn1,1;timn1,1;vrijemen1,1
vozač1,2;tim1,2;vrijeme1,2;vozač2,2;tim2,2;vrijeme2,2;...;vozačn2,2;timn2,2;vrijemen2,2
...
vozač1,k;tim1,k;vrijeme1,k;vozač2,k;tim2,k;vrijeme2,k;...;vozačnk,k;timnk,k;vrijemenk,k
kraj
```

Primjer ulaza je (na ulazu nema razmaka, učitavanje se prekida unosom stringa „kraj”):
*Verstappen;Red Bull Racing;140;Noriss;McLaren;165;Bottas;Alfa Romeo
 Racing;146;Hamilton;Mercedes;200
 Noriss;McLaren;230;Hamilton;Mercedes;210;Verstappen;Red Bull Racing;194
 kraj*

Prvo mjesto na utrci donosi 25 bodova, drugo 18 bodova, treće 15, a zatim 12,10,8, 6,4,2,1 bod do 10. mjesta. Dodavanjem `std::greater<int>` kao zadnji parametar u listi parametara tipa se može postići da se STL-ovi koji su sortirani po tipu `int` sortiraju u silaznom poretku.

- Osmislite i implementirajte (korištenjem STL-a) odgovarajuće strukture za spremanje i računanje rang liste vozača tako da s najmanjim brojem iteriranja po trojkama i vozačima možemo ispisati tablicu (vozači s istim brojem bodova ne moraju imati jednaku poziciju):
 1. mjesto: vozač₁ brojBodova₁
 2. mjesto: vozač₂ brojBodova₂
 3. mjesto: ... itd.
- Osmislite i implementirajte (korištenjem STL-a) odgovarajuću strukturu za spremanje i računanje liste vozača tako da tako da s najmanjim brojem iteriranja po trojkama i vozačima možemo ispisati tablicu:

```
vozač1 brPutaprvi1 brPutadrugi1 brPutatreći1 brPutacetvrti1
```

...

```
vozačk brPutaprvik brPutadrugik brPutatrećik brPutacetvrtik
```

Vozači su sortirani prema četvorkama (brPutaprvi, brPutadrugi, brPutatreći, brPutacetvrti)

Napomena: Smijete se koristiti pisanim materijalima, web-stranicom kolegija i linkovima koji vode s nje. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultirat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretipkavanje teksta zadataka. Smijete koristiti samo svoj korisnički račun na računalu student! **Za rješavanje zadatka morate koristiti STL a ne obična polja!**
Zabranjeno je korištenje biblioteke `<bits/stdc++.h>`

PREZIME I IME:

MATIČNI BROJ STUDENTA:

BODOVI:

Drugi zadatak (15 bodova)

2. grupa

Sezona utrka formule 1 sadrži k utrka. Svaka utrka se pamti kao vektor trojki koje sadrže ime vozača, ime tima i konačno mjesto (poziciju) u utrci. Pretpostavljamo da su sve pozicije različite. Trojku uredite po želji, međutim koristite konzistentno u rješenju zadatka. Ukoliko sa ulaza učitavamo informacije o utrkama na sljedeći način:

```
vozač1,1;tim1,1;pozicija1,1;vozač2,1;tim2,1;pozicija2,1;...;vozačn1,1;timn1,1;pozicijan1,1
vozač1,2;tim1,2;pozicija1,2;vozač2,2;tim2,2;pozicija2,2;...;vozačn2,2;timn2,2;pozicijan2,2
.
.
.
vozač1,k;tim1,k;pozicija1,k;vozač2,k;tim2,k;pozicija2,k;...;vozačnk,k;timnk,k;pozicijank,k
kraj
```

Primjer ulaza je (na ulazu nema razmaka):

```
Verstappen;Red Bull Racing;1;Noriss;McLaren;3;Bottas;Alfa Romeo
Racing;2;Hamilton;Mercedes;4
Noriss;McLaren;3;Hamilton;Mercedes;2;Verstappen;Red Bull Racing;1
kraj
```

Top 10 timova dobiva novčane nagrade u iznosima 60 milijuna \$, 50 milijuna \$, 40 milijuna \$, 30 milijuna \$, 20 milijuna \$, 16 milijuna \$, 15 milijuna \$, 14 milijuna \$, 13 milijuna \$, 12 milijuna \$, dok svi ostali timovi osvajaju po 10 milijuna \$ za sudjelovanje. Dodavanjem `std::greater<int>` kao zadnji parametar u listi parametara tipa se može postići da se STL-ovi koji su sortirani po tipu `int` sortiraju u silaznom poretku.

- a) Osmislite i implementirajte odgovarajuće strukture (korištenjem STL-a) za spremanje i računanje ukupne novčane nagrade koju su osvojili timovi u sezoni tako da s najmanjim brojem iteriranja po trojkama i vozačima možemo ispisati tablicu:


```
tim1 iznos$1
tim2 iznos$2
tim3 iznos$3 ... itd.
```
- c) Ispišite koliko različitih timova je osvojilo jedno od prva tri mjesta na bilo kojoj od k utrka tako da broj iteriranja po trojkama i vozačima bude minimalan.

Napomena: Smijete se koristiti pisanim materijalima, web-stranicom kolegija i linkovima koji vode s nje. Ne smijete se koristiti nikakvim sredstvima komunikacije (papirići, mobitel, mail i sl.). Svaki pokušaj prepisivanja i dogovaranja rezultirat će trenutnim udaljavanjem sa kolokvija. Strogo je zabranjeno fotografiranje ili pretpikavanje teksta zadataka. Smijete koristiti samo svoj korisnički račun na računalu student! **Za rješavanje zadatka morate koristiti STL a ne obična polja!**
Zabranjeno je korištenje biblioteke `<bits/stdc++.h>`

PREZIME I IME:

MATIČNI BROJ STUDENTA:

BODOVI: