

UNIVERZITET U TUZLI
FAKULTET ELEKTROTEHNIKE



Objektno-orijentirano programiranje

Zadaća 3

Tuzla, decembar 2021.

Sadržaj

Zadatak 1	2
Zadatak 2	3
Zadatak 3	3
Zadatak 4	4
Zadatak 5	4

Zadatak 1

Napisati program koji omogućava rad sa matricama. Program treba da omogući operacije sabiranja i množenja matrica, kao i pronalazak elemenata na glavnoj i sporednoj dijagonali.

Neophodno je da korisnik sa standardnog ulaza unese dimenzije za matrice. Elemente u matrici generisati korištenjem *rand()* funkcije u opsegu [0, 100]. Ukoliko nije moguće izvršiti neku od navedenih operacija (množenje, sabiranje ili pronalazak elemenata na glavnoj i sporednoj dijagonali) generisati iznimku tipa `std::logic_error`. Neophodno je koristiti više fajlova (deklaracije funkcija izdvojiti u zaglavlje pod imenom *Matrica.hpp*, definicije funkcija u *Matrica.cpp*). U fajlu *main.cpp* smije postojati samo *main()* funkcija u kojoj pozivamo funkcije:

- Generisanje matrica: *kreirajMatricu()*
- Sabiranje matrica: *saberiMatrice()*
- Množenje matrica: *pomnoziMatrice()*
- Pronalazak elemenata na glavnoj dijagonali: *pronadjiGlavnuDijagonalu()*
- Pronalazak elemenata na sporednoj dijagonali: *pronadjiSporednuDijagonalu()*
- Ispis matrice: *ispisiMatricu()*

Obratiti pažnju na način prosljeđivanja argumenata u funkciju.

Način izvršavanja programa je prikazan na narednoj slici:

<pre>Unesi dimenzije matrice A Broj redova: 4 Broj kolona: 4 Matrica A ----- 17 11 81 93 26 29 4 80 96 94 55 13 66 57 83 31 ----- Unesi dimenzije matrice B Broj redova: 4 Broj kolona: 4 Matrica B ----- 87 5 17 60 73 56 22 72 53 23 17 43 54 35 1 80 ----- Unesi operaciju: + ----- A + B ----- 104 16 98 153 99 85 26 152 149 117 72 56 120 92 84 111 ----- Main diagonal(A) 17 29 55 31 Main diagonal(B) 87 56 17 80 Secondary diagonal(A) 93 4 94 66 Secondary diagonal(B) 60 22 23 54</pre>	<pre>Unesi dimenzije matrice A Broj redova: 5 Broj kolona: 3 Matrica A ----- 64 10 65 18 91 40 43 78 15 72 44 62 22 90 100 ----- Unesi dimenzije matrice B Broj redova: 3 Broj kolona: 5 Matrica B ----- 25 41 76 42 3 36 32 20 3 35 87 34 93 41 31 ----- Unesi operaciju: * ----- A * B ----- 5475 11268 24534 8979 10016 9588 16644 6260 26226 9198 939 11562 657 10955 8742</pre>
---	--

Zadatak 2

Napisati funkciju `saberi` koja može primiti argumente proizvoljnih tipova. Za numeričke tipove ova funkcija vraća rezultat generiran korištenjem `+` operatora, dok za stringove i niz karaktera vrši spajanje (konkatenacija). Funkcija može primiti opcioni (treći argument) koji ako je proslijeđen množi rezultat sabiranja.

Primjer:

`suma(1,2,3)` - rezultat će biti $(1+2)*3$.

Za stringove omogućava višestruko spajanje na sljedeći način:

`("abc","d", 3)` = `"abcdabcdabcd"`.

Treći defaultni argument ako nije prosljeđen treba da ima vrijednost 1.

Zadatak 3

Napisati funkciju **`transformiraj`** koja uzima dva vektora realnih brojeva, te dodatni parametar, `lambda` izraz koji definira transformaciju. Ukoliko su vektori iste dužine, funkcija transformira elemente prvog vektora aplicirajući operaciju definiranu `lambda` izrazom na odgovarajuće elemente oba vektora. U slučaju da se dužine vektora razlikuju, funkcija treba da generira iznimku tipa **`domain_error`**.

Funkcija ne vraća ništa. Izabrati odgovarajući način prosljeđivanja parametara u funkciju. Napisati i odgovarajući program kojim se testira kreirana funkcija.

Zadatak 4

Definirati zaglavlja i implementaciju za sljedeće module:

- `IOrijeci.hpp` Modul za unos i ispis vektora string-ova. Modul treba da ima dvije funkcije jednu za unos drugu za ispis vektora.
- `dodavanje.hpp` Modul koji sadrži dvije funkcije. Prva verzija funkcije uzima string `a`, karakter `b` i cijeli broj `c`. Funkcija dodaje `c` karaktera `b` na lijevoj strani stringa `a`. Druga verzija funkcije uzima string `a` i cijeli broj `c`. Funkcija dodaje `c` karaktera `' '` na lijevoj strani stringa `a`. Koristiti overloading ulaznih parametara.
- `MAXrijec.hpp` Modul sadrži jednu funkciju koja nalazi riječ unutar vektora riječi koja ima maksimalnu dužinu. Funkcija vraća pronađenu riječ.

Za sve funkcije u gornjim modulima odabrati najadekvatniji način za prosljeđivanje parametara.

Napisati glavni program koji koristeći gore navedene module čita niz riječi a zatim ispisuje učitani niz sortiran po dužini riječi i poravnat na desnu stranu kao u sljedećem primjeru pozivanja programa.

```
adfa
gadgdgda
dsf
adfaffadfafa
adf
ds
^Z

      ds
      dsf
      adf
      adfa
    gadgdgda
  adfaffadfafa
```

Zadatak 5

Napisati program koji omogućava kreiranje proizvoljnog broja objekata tipa *Radnik*.

Broj radnika koje treba kreirati je neophodno unijeti sa ulaza.

Radnik je struktura koja u sebi ima sljedeće članove:

- *Ime*
- *Prezime*
- *Godina rođenja*
- *Mjesto stanovanja*
- *Iznos plate*

Sve kreirane radnike je neophodno spremati u *vektor*.

Napisati funkcije za:

- Unos radnika: *unos()*
- Ispis radnika: *ispis()*

Odabrati pravi način za proslijeđivanje argumenata u funkciju. Nakon završenog unosa radnika, neophodno je izvršiti sortiranje radnika na način opisan u nastavku:

- Po plati: Radnik sa najvećom platom treba biti na početku vektora, radnik sa najmanjom platom na kraju.
- Ukoliko postoji više radnika sa jednakom platom, kao dodatni kriterij za sortiranje treba da bude na osnovu imena tako da imena radnika budu sortirana alfabetski.
- Ukoliko postoji više radnika sa jednakom platom i istim imenom, radnike sortirati po godini rođenja tako da radnik koji je stariji bude ispred.

Strukturu *Radnik* neophodno je kreirati u zaglavlju pod nazivom **Radnik.hpp**. Funkcije za unos i ispis radnika definirati u cpp fajlu pod imenom **Radnik.cpp**. U main.cpp fajlu smije postojati **samo** main() funkcija u kojoj koristimo funkcije za unos i ispis radnika.