**Program: Trgovina**

**enum Vrsta** { kupac = 1, trgovac = 2 }; **struct Proizvod** *{ int rednibroj; char naziv[20]; float cijena; int dostupno\_komada; };* **struct Korisnik** *{ int id; char lozinka[20]; char ime[20]; char prezime[20]; float stanje\_racuna; bool popust; Proizvod narudzba[]; Vrsta pozicija; };*

Prilikom pokretanja programa korisniku se nudi da izabere jednu od dvije ponuđene opcije:

**1 – KUPAC 2 – TRGOVAC (traži se unos random koda definisanog kao globalna konstanta)**

Na osnovu izabrane opcije, od korisnika se zahtjeva da unese svoj korisnički ID, te nakon toga svoju lozinku (podaci o svim korisnicima su smješteni u datoteci korisnici.txt) U zavisnosti od odabrane opcije korisniku se prikazuju sljedeći meniji, gdje svaka od opcija predstavlja zasebnu funkciju (parametri se po potrebi mogu naknadno dodavati)

**MENU 1:**

1. *stanje\_na\_racunu();*
2. *dodaj\_novac\_na\_racun();*
3. *naruci\_proizvod();*
4. *pregled\_narudzbe();*
5. *IZLAZ*

**MENU 2:**

1. *lista\_svih\_proizvoda();*
2. *stanje\_u\_kasi();*
3. *dodaj\_proizvod();*
4. *ukloni\_proizvod();*
5. *promijeni\_cijenu\_proizvoda();*
6. *IZLAZ*

**Funkcije:**

**stanje\_na\_racunu();** - provjerava i ispisuje koliko korisnik ima novca na računu

**dodaj\_novac\_na\_racun();** - korisnik može dodati određeni iznos na svoj račun u intervalu od 10 do 100

**naruci\_proizvod();** - korisniku se ispisuju svi dostupni proizvodi, nakon unosa rednog broja proizvoda provjerava se da li korisnik ima dovoljno novca na računu. Poslije odabranog proizvoda, nudi mu se opcija da nastavi s narudžbom (ili se može dodati da korisnik unese željenu količinu za jedan proizvod), gdje mu se za svaki naredni proizvod daje popust na cijenu od 10%. Na kraju se smanjuje broj dostupnih proizvoda i korisniku se skida novac s računa, a taj isti iznos se dodaje u kasu trgovine!

**pregled\_narudzbe();** - ispisuje sve proizvode koje je korisnik naručio, na osnovu datoteke narudzbe.txt

**lista\_svih\_proizvoda();** – ispisuje listu svih proizvoda, koji su spremljeni u datoteci proizvodi.txt

**stanje\_u\_kasi();** - ispisuje stanje u kasi na osnovu podataka iz datoteke kasa.txt

**dodaj\_proizvod();** - dodavanje novog proizvoda s postavljanjem naziva, cijene i broja dostupnih komada

**ukloni\_proizvod();** - uklanja proizvod na osnovu unesenog rednog broja

**promijeni\_cijenu\_proizvoda();** - pomoću prosljeđenog rednog broja, korisnik postavlja novu cijenu za željeni proizvod

**Dodatne napomene:**

* Za svaki unos dodati provjeru da li unesena vrijednost postoji u odgovarajućoj datoteci i da li je zadovoljen traženi interval, ako ne ispisati odgovarajuću poruku (jedinstveno definisana greska(); funkcija)
* Za svaku vrstu pogrešnog unosa u odnosu na tip promjenjive, dodati ponavljanje unosa dok se ne unese zahtjevani tip podatka
* Sve funckije prvo deklarisati iznad main-a, a ispod main-a ih implementirati
* Koristiti komentare na početku programa u obliku /\*=========\*/ (više redova), a na početku svake funkcije u obliku //=========, te na nekim mjestima u funkciji koja zahtjevaju dodatno objašnjenje
* Ispis formatirati korištenjem tabulatora i naredbi iz biblioteke <iomanip>, te razmaka između svakog ispisa u obliku ------------------- (jedinstveno definisana linija(); funkcija)
* Koristiti naredbe za brisanje terminala nakon svakog unosa / ispisa (system(„cls“), te umjesto standarde naredbe system(„pause“) koristiti funkciju getch() iz biblioteke <conio.h> koja će se pozivati nakon poruke „Za nastavak pritisnite bilo koju tipku...“
* Po mogućnosti implementirati po jednu rekurzivnu i lambda funkciju