

DEO II – PYTHON PROGRAMIRANJE

Praktični deo ispita traje **maksimalno 150 minuta**.

Student od ponuđenih pet zadataka bira četiri koje radi.

Za vreme trajanja ispita nije dozvoljeno korišćenje mobilnih telefona ili drugih komunikacionih uređaja, i nije dozvoljen pristup Internetu.

Student je položio ispit ako položi oba dela, tj. da ima najmanje 5 poena na teorijskom delu, i najmanje 10 poena na praktičnom delu ispita.

Student koji prepisuje na ispitu, udaljava se sa ispita.

Uputstvo za imenovanje zadataka:

Svaki zadatak čuvati u direktorijumu **CS100 broj_indeksa**, gde je **broj_indeksa** Vaš broj indeksa. Datoteke sa .py ekstenzijom naziovite **CS100_broj_indeksa_zadatak_no.py**, gde je **broj_indeksa** Vaš broj indeksa, a **zadatak_no** je broj zadatka.

Primer:

```
\CS100 1234\  
└──CS100_1234_zadatak_01.py  
└──CS100_1234_zadatak_02.py  
└──CS100_1234_zadatak_03.py  
└──CS100_1234_zadatak_04.py
```

ZADACI:

1. Taksista u svojoj dnevnoj smeni vozi 20 vožnji. Svaka vožnja različito košta: minimalna cena vožnje je 200 RSD, a maksimalna cena vožnje je 2500 RSD.

Izračunati koliko novca taksista zaradi za 10 radnih dana. Cenu vožnje generisati nasumično koristeći funkciju **randint(a, b)** iz modula **random**.

2. Napisati funkciju **print_numbers_to_file(file_name, n)**, koja otvara datoteku **file_name.txt** i u svakom redu ispisuje po jedan broj: $10^{-\frac{n}{2}}$, $10^{-\frac{n}{2}-1}$, ..., $10^{\frac{n}{2}-2}$, $10^{\frac{n}{2}-1}$. Funkcija treba da radi i ukoliko korisnik unese ekstenziju datoteke, ali i ako ne unese.

U glavnom programu otvoriti datoteku **file_name.txt**, učitati svaki broj manji od 1 i prikazati zbir tih brojeva. Obraditi izuzetak za nepostojeću datoteku pisanjem poruke „**Datoteka nije pronađena**“. Program treba u svakom slučaju da ispiše „**Kraj programa**“, čak i ako uhvati izuzetak.

Koristiti preporuke za pisanje koda u Python jeziku.

Primer:

```
print_numbers_to_file('zadatak2', 3)
```

$$10^{-\frac{3}{2}}, 10^{-\frac{2}{2}}, 10^{-\frac{1}{2}}, 10^{-\frac{0}{2}}, 10^{\frac{1}{2}}, 10^{\frac{2}{2}}$$

zadatak2.txt:

```
0.03162277660168379
0.1
0.31622776601683794
1
3.1622776601683795
10
```

Izlaz glavnog programa -> 0.44785054261852175

3. Napisati rekurzivnu funkciju **pronadji_velika_slova(tekst)** koja vraća broj velikih slova u prosleđenom tekstu. Zatim napisati rekurzivnu funkciju **pronadji_mala_slova(tekst)** koja vraća broj malih slova u prosleđenom tekstu. U glavnom programu pronaći razliku između velikih i malih slova u tekstu koji unosi korisnik i prikazati rezultat na konzoli.

Zabranjeno je korišćenje globalnih promenljivih. Koristiti preporuke za pisanje koda u Python jeziku.

Primer:

```
pronadji_velika_slova("TesTiRamO") -> 4
pronadji_mala_slova("TesTiRamO") -> 5
razlika -> 4 - 5 = -1
```

4. Napisati funkciju **enter_student_data_and_course()** pomoću koje korisnik unosi sa tastature ime, prezime, broj indeksa, šifru jednog predmeta i naziv tog predmeta. Funkcija vraća listu **my_student**. U glavnom programu pozvati funkciju, i štampati podatke na konzoli, tako da je svaki element liste **my_student** u posebnom redu.

Zabranjeno je korišćenje globalnih promenljivih. Koristiti preporuke za pisanje koda u Python jeziku.

5. Korisnik ima ulogu studentske službe koja za podatke studenta i predmeta unosi predispitne obaveze. Korisnik najpre uvozi funkciju **enter_student_data_and_course()** iz četvrtog zadatka, poziva je zbog unosa podatka studenta, pa sa tastature unosi poene sa projektnog zadatka i ostalih predispitnih obaveza.

Ukoliko student nema 15 poena na projektnom zadatku, program ispisuje poruku da student (ime, prezime, broj_indeksa) nema dovoljno poena na projektnom zadatku iz predmeta (šifra, naziv predmeta). Ukoliko ima dovoljno poena na projektnom zadatku, proverava se da li zbir poena na projektnom zadatku i ostalim predispitnim obavezama iznosi minimalno 35, i tada program ispisuje da student (ime, prezime, broj_indeksa) može izaći na ispit (šifra, naziv predmeta), a ukoliko nema, izlaz javlja da student ne može izaći na ispit.

Ukoliko je broj poena na projektnom zadatku veći od 30, ili broj poena na ostalim predispitnim obavezama veći od 40, ispitati poruku „Greška!“.

Koristiti samo kontrolne strukture.
