



Predmet: CS100 – Uvod u programiranje (Python)

Dan ispita:

IME I PREZIME _____ **BROJ INDEKSA** _____

Pažljivo pročitati sledeće instrukcije.

Prekršaj ovih pravila kao i svaki pokušaj prepisivanja biće sankcionisan udaljavanjem studenta sa ispita i pokretanjem disciplinskog postupka u skladu sa pravilima Univerziteta.

1. Ispit se radi se u okviru ispitnih rokova u učionici Fakulteta.
2. Ispit traje najviše **tri sata** i sastoji se iz *teoretskog dela* (bez korišćenja literature) i *praktičnog dela* (takođe bez korišćenja literature).
3. *Teoretski deo ispita* podrazumeva da student odgovori na postavljena pitanja, radi se na papiru i traje **maksimalno 30 minuta**.
4. *Praktični deo ispita* se sastoji od Python programiranja, radi se na računaru.
5. Prvo se radi teoretski deo ispita, nakon toga praktični deo ispita. Nakon završetka izrade teoretskog dela, student može da pristupi izradi praktičnog dela ispita.
6. Praktični deo ispita traje **maksimalno 150 minuta**. Student od ponuđenih pet zadataka bira četiri koje radi.
7. Student je položio ispit ukoliko osvoji minimum 50% poena na teoretskom delu i minimum 50% poena na praktičnom delu ispita.
8. Za vreme ispita nije dozvoljeno korišćenje mobilnih telefona ili drugih komunikacionih uređaja, i nije dozvoljena poseta bilo kom sajtu koji omogućava komunikaciju (webmail, facebook ili druge društvene mreže, i sl.).

Proble m	Max. Poena	Dobijeni poeni	Proble m	Max. Poena	Dobijeni poeni
P1	2		Z1	5	
P2	2		Z2	5	
P3	2		Z3	5	
P4	2		Z4	5	
P5	2				
Teorija	10		Zadaci	20	

Ukupno:

--

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. Pekar svakog sata proda 25 porcija bureka. Svaka porcija različito košta: minimalna cena je 150 RSD, a maskimalna cena je 1250 RSD.

Izračunati koliko novca pekar zaradi za 2 smene (16 radnih sati). Cenu konkretne porcije generisati nasumično koristeći funkciju **randint(a, b)** iz modula **random**.

2. Napisati funkciju **print_numbers_to_file(file_name, a, b, n)**, koja otvara datoteku **file_name.txt** i u svakom redu ispisuje po jedan broj: $\frac{2}{(a+b)^{-2n}} + \frac{2}{(a+b)^{-n}} + \frac{2}{(a+b)^n} + \frac{2}{(a+b)^{2n}}$

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. Pazite na nulu ispod razlomačke crte!

Primer:

```
print_numbers_to_file('zadatak2', 1,1,1)
```

zadatak2.txt:

```
8
4
2
0,5
```

Izlaz glavnog programa -> 0.5

3. Napisati rekursivnu funkciju **najveci_element(niz)** koja vraća najveći element u prosleđenom nizu brojeva. Zatim napisati rekursivnu funkciju **najmanji_element(tekst)** koja vraća najmanji element u istom prosleđenom tekstu. U glavnom programu napraviti niz od kvadrata brojeva između najmanjeg i najvećeg elementa u nizu i prikazati rezultat na konzoli. Niz unosi korisnik. Koristiti preporuke za pisanje koda u Python jeziku.

Primer:

```
niz = [2, 5, 62, 5, 42, 52, 48, 5]
najveci_element(niz) -> 52
najmanji_element(niz) -> 2
rezultat -> [4,...,2,704]
```

4. Napisati funkciju **enter_financial_data()** pomoću koje korisnik unosi sa tastature ime i prezime kupca, količinu novca na računu, šifru akcije, naziv akcije, broj deonica, kao i da li kupuje ili prodaje. Funkcija vraća listu **my_stocks**.

U glavnom programu pozvati funkciju, i štampati podatke na konzoli, tako da je svaki element liste **my_stocks** u posebnom redu.

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

5. Nacrtati dijagram toka i napisati kod za sledeći zadatak. Menadžer je odlučio da uradi promociju marketa tako što će omogućiti sniženje proizvoda do čak 40% u zavisnosti od količine kupljenih proizvoda. Pravila za popust su sledeća:

Na količinu 10-19 komada dobija se 10% popusta.

Na količinu 20-49 komada dobija se 20% popusta.

Na količinu 50-99 komada dobija se 30% popusta.

Na količinu 100 ili više komada dobija se 40% popusta.

Program od korisnika zahteva da unese količinu proizvoda koji kupuje, a program obračunava koliki popust je korisnik ostvario prilikom kupovine. Korisnik može i da ne ostvari popust ukoliko kupi manje od 10 komada proizvoda, a o tome ga obaveštava program.

Ukoliko je količina proizvoda negativan broj ispitati poruku „Greška!“. Koristiti samo kontrolne strukture.
