

# Metodika nastave računarstva A 4L, Ispit - JUN1

11.06.2020.

Pročitati sve zadatke **pažljivo** pre rada - sve što nije navedeno ne mora da se implementira!  
Napraviti na **Desktop**-u direktorijum sa imenom `mnra_jun1_Ime_Prezime_m1GGXXX` u kome se nalazi vaš projekat.  
Koristiti editor po želji (npr. Kate ili Gedit). Sve kodove ostaviti unutar vašeg direktorijuma.  
Recimo da želite da pokrenete program `1.py`, tada treba da se u terminalu pozicionirajte na vaš direktorijum i da unesete naredbu `python 1.py`. Sa naredbom `pwd` mozete da vidite u kom direktorijumu se trenutno nalazite. Sa naredbom `cd imeDirektorijuma` pozicionirate sa na poddirektorijum `imeDirektorijuma`. Ako u naredbi `cd` umesto `imeDirektorijuma` stavite `..` onda se pozicionirate na roditeljski direktorijum. Sa naredbom `ls` možete da vidite sadržaj direktorijuma na kome se trenutno nalazite. Vreme za rad: **2h. SREĆNO!**

## 1. Binaran zapis (8p)

Napraviti program koji za prirodan broj, koji se unosi sa standardnog ulaza, ispisuje njegov zapis u osnovi 2 na standardni izlaz.

### Primer 1

*ulaz:* 10

*izlaz:* 1010

### Primer 2

*ulaz:* 19

*izlaz:* 10011

### Primer 3

*ulaz:* 0

*izlaz:* 0

## 2. Parcijalne sume (7p)

Napisati rekurzivnu funkciju `parcijalne_sume(lista, n)`, koja za datu listu  $[a_0, a_1, a_2, \dots, a_{n-1}]$  duzine  $n$ , kao rezultat vraća listu  $[a_0, a_0 + a_1, a_0 + a_1 + a_2, \dots, \sum_{i=0}^{n-1} a_i]$ . Zatim napisati program, koji za listu koja se unosi preko standardnog ulaza, ispisuje rezultat primene funkcije `parcijalne_sume(lista, n)` na standardni izlaz.

### Test primer:

*ulaz:*

Unesite n: 5

Unesite elemente liste: 1 2 3 4 5

*izlaz:*

Parcijalne sume: 1 3 6 10 15

## 3. Studenti (8p)

U fajlu `student.py` napravi klasu `student` sa sledećim metodama:

- `__init__(self, ime, prezime)`

Konstruktor za klasu, koji inicializuje odgovarajuća polja objekta. Argumenti `ime` i `prezime` predstavljaju ime i prezime studenta. Osim ovih polja, `student` ima i polje `ocene`, koje predstavljaju listu svih njegovih ocena. Pri konstrukciji, ta lista je prazna.

- `getIme(self)`

Ovaj metod vraća ime studenta.

- `getPrezime(self)`

Ovaj metod vraća prezime studenta.

- `dodajOcenу(self, ocena)`

Ovaj metod dodaje novu ocenu `ocena` u listu studentovih ocena.

- `prosek(self)`

Ovaj metod računa i vraća prosek studenta.

————— Okrenite stranu! —————

- ispisi(*self*)

Ovaj metod ispisuje ime studenta, prezime studenta i njegov prosek u obliku: *Ime Prezime Prosek*

Napraviti program koji će da računa proseke studenata. Koristiti fajl student.py kao moduo. Korisnik preko standardnog ulaza unosi broj  $n$ , pa zatim  $n$  linija, pri čemu se u svakoj liniji nalazi ime, prezime i ocena nekog studenta, separatisani sa space-om. Kao rezultat program treba da ispiše ime, prezime i prosek svakog studenta, sortirane po proseku opadajuće.

**Test primer**

**ulaz:**

Unesite n: 10

Unesite podatke o studentima:

Petar Peric 10

Petar Peric 8

Ana Antic 10

Ana Antic 9

Filip Filipovic 10

Ana Antic 9

Filip Filipovic 10

Ana Antic 10

Marija Maric 9

Stefan Stefanovic 7

**izlaz:**

Filip Filipovic 10

Ana Antic 9.5

Petar Peric 9

Marija Maric 9

Stefan Stefanovic 7

**4. Statistika (7p)**

Napraviti program koji prebrojava broj pojavljivanja svake reči u fajlu. Korisnik unosi ime fajla preko standardnog ulaza. Rezultat ispisati na standardni izlaz.

**Test primer**

*sadržaj fajla ulaz.txt:*

I pocnem da letim, i letim, i letim I letim, i letim, i letim, letim, letim. Pa se spustim, cisto da pokupim hranu. I pice i ribe, i ribe, ribe, ribe.

*ulaz:* ulaz.txt

*izlaz:*

I 3

pocnem 1

da 2

letim, 6

i 6

letim 1

letim. 1

Pa 1

se 1

spustim, 1

cisto 1

pokupim 1

hranu. 1

pice 1

ribe, 3

ribe. 1