

# Metodika nastave računarstva A 4L, Ispit - SEPT1

20.08.2020.

Pročitati sve zadatke **pažljivo** pre rada - sve što nije navedeno ne mora da se implementira!  
Napraviti na **Desktop**-u direktorijum sa imenom `mnra_sept1_Ime_Prezime_m1GGXXX` u kome se nalazi vaš projekat.  
Koristiti editor po želji (npr. Kate ili Gedit). Sve kodove ostaviti unutar vašeg direktorijuma.  
Da bi pokrenuli program, treba da se u terminalu pozicionirate na vaš direktorijum. Recimo da želite da pokrenete program `1.py`, onda treba da ukucate `python 1.py`. Sa naredbom **pwd** mozete da vidite u kom direktorijumu se trenutno nalazite. Sa naredbom **cd imeDirektorijuma** postavljate sa na poddirektorijum imeDirektorijuma. Ako u naredbi **cd** umesto imeDirektorijuma stavite `..` onda se pozicionirate na roditeljski direktorijum. Sa naredbom **ls** mozete da vidite sadržaj direktorijuma na kome se trenutno nalazite. Vreme za rad: **2h. SREĆNO!**

## 1. Dva minimuma (8p)

Napraviti program koji za dati niz brojeva  $a_1, a_2, \dots, a_n$  pronalazi dva najmanja člana niza. Preko standardnog ulaza, korisnik prvo unosi broj elemenata niza  $n$ , zatim i elemente niza. Na standardni izlaz ispisati vrednosti dva najmanja člana niza. U slučaju da niz nema barem dva elementa, ispisati odgovarajuću poruku. Voditi računa o složenosti vašeg programa. U slučaju da je složenost vašeg koda nije dobra, možete dobiti najviše 50% poena.

Test primeri

- **Ulaz:**

Unesite n: 4

4 2 3 1

- **Izlaz:**

1 2

- **Ulaz:**

Unesite n: 5

1 2 1 3 1

- **Izlaz:**

1 1

- **Ulaz:**

Unesite n: 1

- **Izlaz:**

Niz mora da ima barem dva elementa.

## 2. $e^x$ (7p)

Napisati rekurzivnu funkciju koja računa  $n$ -tu parcijalnu sumu Tejlorovog reda funkcije  $e^x$ .

Tejlorov red:  $\sum_{n=0}^{\infty} x^n/n! = e^x$ .

$n$ -ta parcijalna suma:  $\sum_{i=0}^n x^i/i!$

Korisnik preko standardnog ulaza unosi realan broj  $x$  i prirodan broj  $n$ . Na standardni izlaz ispisati vrednost  $n$ -te parcijalne sume Tejlorovog reda u tački  $x$ .

Test primeri:

- Unesite n: 2

Unesite x: 1

Rezultat: 2.5

- Unesite n: 3

Unesite x: 1.5

Rezultat: 4.1875

- Unesite n: 4

Unesite x: -2

Rezultat: 0.3333333

### 3. Ciklične grupe (8p)

U fajlu algebra.py napraviti klasu *CiklicnaGrupa* sa sledećim metodama:

- `__init__(self, n)`  
koji predstavlja konstruktor klase. Klasa sadrži samo jedan atribut  $n$ . Ako je  $n = 0$ , smatrati da formiramo grupu celih brojeva  $Z$ . Inače, formiramo grupu  $Z_n$ , sa elementima  $0, 1, 2, \dots, n - 1$ .
- `zbir(self, x, y)`  
Ovaj metod vraća zbir elemenata u odgovarajućoj cikličnoj grupi. Ako je  $n = 0$ , vratiti standardni zbir  $x + y$ . Inače vratiti  $x +_n y$ .
- `inverzZaSabiranje(self, x)`  
Ako je  $n = 0$ , vratiti inverz za sabiranje u grupi  $Z$ . Inače vratiti inverz za sabiranje po modulu  $n$ , tj.  $y$  t.d.  $x +_n y = 0$ .
- `inverzZaMnozenje(self, x)`  
Ako je element inverzibilan u cikličnoj grupi, vratiti uređen par  $(True, x^{-1})$ , gde je  $x^{-1}$ , inverz datog elementa  $x$ , tj. važi da je  $x * x^{-1} = 1$ . Ako element nije inverzibilan, vratiti  $(False, None)$ . U cikličnoj grupi  $Z$ , jedini inverzibilni elemneti su 1 i  $-1$ . U cikličnim grupama  $Z_n$  jedini inverzibilni elementi su oni elementi  $x$  za koje važi da su uzajamno prosti sa  $n$ .
- `razlika(self, x, y)`  
Ovaj metod vraća razliku elemenata u odgovarajućoj cikličnoj grupi. Ako je  $n = 0$ , vratiti standardnu razliku  $x - y$ . Inače vratiti  $x -_n y$ . *Hint*: razliku možete formirati preko metoda `saberi` i `inverzZaSabiranje`.

U fajlu 3.py koristiti fajl algebra.py kao moduo. Korisnik sa standardnog ulaza unosi broj  $n$ . Formirati objekat klase *CiklicnaGrupa* za dato  $n$ . Zatim korisnik unosi dva broja  $x$  i  $y$ . Za unete brojeve izračunati:  $x + y$ ,  $x - y$ ,  $x^{-1}$  i  $y^{-1}$ , koristeći metode klase *CiklicnaGrupa* i ispisati rezultate na standardni izlaz.

### 4. Spajanje (7p)

Napraviti program koji naizmenično kopira liniju po liniju dva fajla u treći fajl. Korisnik preko standardnog ulaza unosi imena dva fajla za kopiranje. Rezultat smestiti u fajl `izlaz.txt`. Ako fajlovi nemaju isti broj linija, višak linija većeg fajla prekopirati u rezultujući fajl.

#### Test primer

ulaz1.txt:

Moja prva linija.  
Moja druga linija.  
Moja treca linija.

ulaz2.txt:

Linija 1.  
Linija 2.  
Linija 3.  
Linija 4.  
Linija 5.

izlaz.txt:

Moja prva linija.  
Linija 1.  
Moja druga linija.  
Linija 2.  
Moja treca linija.  
Linija 3.  
Linija 4.  
Linija 5.