

Prvi računarski kolokvijum

Napomene:

1. Za svaki zadatak kreirati zasebnu skriptu pod nazivom zadatakN.py gde N označava redni broj zadatka. Nije dozvoljeno korišćenje Python funkcija za rešavanje sistema jednačina, aproksimaciju funkcija, određivanje nula funkcija, traženje minimuma i maksimuma.
2. Sve zadatke snimiti u .zip arhivu pod nazivom **SMERXZ_YYYY_Ime_Prezime**, gde je SMER šifra smeru, XZ broj indeksa studenta, YYYY godina upisa (**NE KORISTITI ĆIRILIČNE NITI DIJAKRITIČKE LATINIČNE ZNAKOVE kao što su Č, Ć, Đ, Š, Ž i slično!**). npr. RA1_2011_Imenko_Prezimenovic.zip. Snimljenu arhivu poslati putem [forme](#). Rok za slanje rešenja je petak, 10. decembar do 17:00h.

1. Pokazati da je $\sqrt{a_0} - \frac{2}{5}a_1 - a_2 + a_3^2 = 1$ ako je poznato da funkcija $P(t) = a_0 + a_1t + a_2t^2 + a_3t^3$ zadovoljava sledeće uslove:

$$P(x_1) = 5.625$$

$$P(4x_1 - 2x_2) = -0.512$$

$$P(2x_1) = x_2 + 14.3$$

$$P(x_2) = 44.469$$

$$x_2 - x_1 = 1.2$$

Vrednost konstante x_2 je 1.7. Zadatak rešiti upotrebom proizvoljne metode za rešavanje sistema linearnih jednačina.

2. Date su funkcije:

$$f(x) = \sin(x^2 - 2)$$

$$g(x) = x \cos(x)$$

- a) Nacrtati grafike obe funkcije i x-osu na intervalu $x \in [-3, 1]$.
- b) Proizvoljnom zatvorenom metodom pronaći i nacrtati nule funkcije f na grafiku pod a).
- c) Proizvoljnom zatvorenom metodom pronaći i nacrtati sve preseke funkcije f i g i označiti ih na grafiku pod b).
- d) Odrediti maksimum funkcije f i minimum funkcije g na zadatom intervalu. Označiti pronađene tačke na grafiku pod c) i ispisati njihove vrednosti u konzoli
- e) Odrediti presečne tačke funkcije f i g sa pravom $x = \frac{\pi}{6}$. Ispisati ih u konzoli i označiti ih na grafiku pod d).

3. Data je funkcija:

$$f(x) = \sqrt{2} \sin(2x) + \cos\left(-\frac{x}{3}\right)$$

- Nacrtati funkciju $f(x)$ na intervalu $x \in \left[-\frac{\pi}{2}, \frac{4\pi}{3}\right]$.
- Odrediti potreban broj uniformno (jednako) raspoređenih tačaka za interpolaciju polinomom 4. stepena i označiti ih na grafiku pod a).
- Aproksimirati datu funkciju polinomom 4. stepena na istom intervalu primenom Lagranžove interpolacije. Ispisati dobijeni polinom u konzoli i nacrtati ga crvenom bojom na grafiku pod b).
- Generisati 100 uniformno raspoređenih tačaka na početnom intervalu i aproksimirati funkciju polinomom 3. stepena. Ispisati dobijeni polinom u konzoli i nacrtati ga braon bojom na grafiku pod c).
- Naći presečne tačke polinoma 4. stepena i polinoma 3. stepena, ispisati njihove vrednosti i nacrtati ih na grafiku pod d).

4. Napisati sledeće Python funkcije:

- fz4a – funkcija koja prihvata matricu A proizvoljnih dimenzija m i n , i kvadratnu matricu B dimenzije k , tako da je $k < m$ i $k < n$. Za svaku kvadratnu podmatricu M matrice A dimenzije k izračunati sumu jediničnog proizvoda matrica M i B , i rezultat smestiti u odgovarajuće mesto povratne matrice kao na slici.

1	2	3		4	3		13	0
0	3	7		2	1		0	0
4	1	2						

1	2	3		4	3		13	30
0	3	7		2	1		0	0
4	1	2						

1	2	3		4	3		13	30
0	3	7		2	1		18	0
4	1	2						

1	2	3		4	3		13	30
0	3	7		2	1		18	37
4	1	2						

Primeri:

```
A = np.array([
    [1, 2, 3],
    [0, 3, 7],
    [4, 1, 2]
])
B = np.array([
    [4, 3],
    [2, 1]
])
Rezultat:
[[13. 30.]
 [18. 37.]]
```

```
A = np.array([
    [1, 2, 3, 1, 1],
    [0, 3, 0, 1, 3],
    [0, 3, 0, 1, 4],
    [0, 3, 2, 1, 2],
    [4, 1, 2, 1, 4]
])
B = np.array([
    [0, 3, 1],
    [0, 2, 2],
    [2, 1, 1]
])
Rezultat:
[[18. 19. 17.]
 [20. 12. 23.]
 [30. 12. 22.]]
```

- b) **fz4b** – funkcija koja vraća *True* ukoliko se u bilo kojem redu, koloni ili dijagonali ulazne matrice proizvoljnih dimenzija n i m ($n, m > 2$) nalaze tri ista uzastopna broja, a u suprotnom vraća *False*.

Primeri:

```
A = np.array([
    [1, 2, 1, 1],
    [0, 0, 5, 3],
    [1, 4, 1, 4],
    [0, 0, 7, 1]
])
```

Rezultat: False

```
A = np.array([
    [1, 2, 1, 4],
    [0, 0, 5, 3],
    [1, 4, 1, 4],
    [0, 0, 0, 1]
])
```

Rezultat: True

```
A = np.array([
    [1, 2, 1, 4],
    [0, 0, 1, 3],
    [1, 4, 1, 4],
    [0, 0, 7, 1]
])
```

Rezultat: True

```
A = np.array([
    [1, 2, 1, 4],
    [0, 0, 4, 3],
    [1, 4, 1, 4],
    [0, 0, 7, 1]
])
```

Rezultat: True