**Uvod u programiranje, Test broj 2**

1. **(1 poen)** Sta je rezultat izvrsavanja programa

# include <stdio.h>

main() {

double k = 0;

for (k = 0.0; k < 3.0; k++);

printf("%lf", k);

}

2. **(1 poen)** Sta je rezultat izvrsavanja programa

#include <stdio.h>

main() {

int i = 0;

while (i < 9) {

i++;

printf("Laza\n");

while (i < 6) {

i++;

printf("Pera\n");

}

}

}

3. **(1 poen)** Sta je rezultat izvrsavanja programa (ako je ucitano 2)

#include <stdio.h>

main() {

double zn;

printf("Unesite broj izmedju 1 i 2: ");

scanf("%lf", &zn);

switch (zn) {

case 1:

printf("1\n");

break;

case 2:

printf("2\n");

break;

}

}

4. **(2 poena)** Sta je definisano sledecom definicijom?

float (\*b)();

5. **(1 poen)** Fibonacijev niz je niz brojeva definisan na sledeci nacin:

f[1] = 1, f[2] = 1, f[n] = f[n-1] + f[n-2] (za n > 2)

Napisati funkciju int nTi(int n) koja odredjuje vrednost n-tog elementa niza f. **Ne koristiti nizove kao vrstu podataka**.

6. **(2 poena)** Neka je *a* broj veci od nule. Tada jednacina  ima jedno resenje koje se racuna tako sto se formira niz brojeva:

 ,  (za *n* > 1)

Postupak se prekida onog trenutka kada apsolutna razlika dva poslednja sracunata elementa postane manja od zadatog broja *eps* (kazemo da je tada pronadjeno resenje sa tacnoscu *eps*) i kao resenje se uzima poslednji izracunati element. Napisati funkciju double resenje(double a, double eps, int m) koja za zadate vrednosti *a* i *eps* odredjuje i vraca priblizno resenje jednacine  sa tacnoscu *eps*, ali tako da ako nakon izarcunavanja elementa *xm* jos uvek nije dostignuta tacnost, prekida dalje racunanje i vraca element *xm*. U funkciji **ne korisititi nizove**.

7. **(2 poena)** Napisati funkciju int drugiNajveci(int n, int a[]) koja pronalazi i vraca vrednost drugog najveceg elementa nniza a. Pretpostaviti da u nizu nema jednakih elemenata.