**Ime:**

**Prezime :**

**Matični broj:**

1. Za idući kôd odrediti vremensku složenost T(N) i vremensku klasu složenosti O(f(N)) (10 bodova):

int f(int \*niz, int n) {

int i = 0, sum = 0;

while(i < n \* n) {

int j = 0;

while (j < 2 \* n) {

sum += niz[j];

j++;

}

i++;

}

return sum;

}

1. Procijeniti složenost O() algoritma koji računa operaciju za skup (niz) i niz brojeva, dužine N i M. Algoritam iz prvog niza ispisuje element koji se u drugom nizu pojavljuje najviše puta. Algoritam se može napisati na dva načina:
2. Nijedan niz nije sortiran. Prolazimo kroz prvi niz i za svaki broj brojimo koliko puta se pojavljuje u drugom nizu.
3. Sortiramo oba niza uzlazno. Zatim za svaki broj prvog niza brojim u drugom nizu koliko puta se pojavljuje bez vraćanja.

Složenost izraziti u N i M. Možete pretpostaviti da se za sortiranje u (b) koristi quicksort.

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj) i jedan broj. Funkcija će izbrojati koliko je elemenata liste veće od tog broja. (25 bodova)
2. Napisati funkciju koja prima vezanu listu (svaki element sadrži broj) i jedan dodatni broj. Funkcija će dodati novi element sa tim brojem na sredinu liste (ili odmah nakon sredine ako je broj elemenata neparan). (35 bodova)
3. Prikazati quicksort algoritam na nizu { 8, 4, 12, 3, 10, 7, 3, 6, 9 }. (10 bodova)