**Ime:**

**Prezime :**

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) {

sum += niz[i];

n--;

}

return sum;

}

1. Procijeniti složenost O() algoritma koji računa operaciju za skup (niz) i niz brojeva, dužine N i M. Algoritam zbraja sve elemente iz prvog niza koji su manji od svih brojeva u drugom nizu. Algoritam se može napisati na tri načina:
2. Nijedan niz nije sortiran. Prolazimo kroz prvi niz i za svaki broj provjeravamo da je manji od svih elemenata u drugom nizu.
3. Sortiramo drugi niz uzlazno i zatim zbrajamo sve brojeve koji su manji od prvog broja u drugom nizu.
4. Sortiramo oba niza uzlazno. Zatim zbrajamo sve brojeve prvog niza dok ne dođemo do broja koji je veći ili jednak od prvog broja u drugom nizu.

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

(20 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj). Funkcija će izbrojati koliko je brojeva liste veće od prethodnog broja u listi. (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 ispred prvog broja u listi koji je veći od 100 (ako takav postoji). (35 bodova)
3. Prikazati quicksort algoritam na nizu { 7, 3, 11, 2, 9, 6, 3, 8, 9 }. (10 bodova)