**Ime Prezime:**

1. Za idući kôd odrediti T(N) i O(N) (15 bodova):

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

int i = n, sum = 0;

while(i > 0) {

int j = 0;

while(j < i) {

sum += niz[j];

j++;

}

while(j < n) {

sum += niz[j];

j++;

}

i--;

}

return sum;

}

1. Procijeniti klasu složenosti O() algoritma koji u nizu pozitivnih brojeva (dužine N) traži dva broja čija razlika daje 0. Algoritam samo vraća True ili False ovisno o tome da li postoje takva dva broja u nizu. Algoritam se može napisati na dva načina:
2. Niz nije sortiran. Svaki broj u nizu pokušamo oduzeti od svakog od preostalih brojeva u nizu.
3. Niz je sortiran. Postavimo dva indeksa na početak i kraj niza (L i R). Ako je razlika brojeva na ta dva indeksa veća od 0, umanjujemo R. Ako je razlika manja od nula, uvećavamo L. Algoritam se prekida ako L i R postanu jednaki.

(15 bodova)

1. Napisati funkciju koja prima jednostruko vezanu listu (svaki element sadrži broj b). Funkcija vraća broj elemenata liste čiji su prethodni i idući element jednaki (brojevi su im jednaki). Kolika je složenost funkcije O()? (25 bodova)
2. Napisati funkciju koja prima listu i broj. Funkcija u listu dodaje novi element sa brojem 0 ispred svakog elementa sa pozitivnim brojem u listi. Funkcija vraća izmijenjenu listu. Kolika je složenost funkcije O()? (35 bodova)
3. Prikazati quicksort algoritam na nizu { 8, 5, 6, 1, 1, 7, 2 }. (10 bodova)