



Ime i prezime: _____

Broj indeksa: _____

UPUTSTVO ZA RAD

- Uredno popunite svoje podatke na papiru. Na desktopu napravite folder oblika **arh_kol_ImePrezime_mrGGBBB**.
- Ispit se sastoji od 2 zadatka. Oba zadatka nose po 25 bodova.
- Studenti koji zadržavaju poene sa Intel-a, rade samo Arm. Potrebno je da naglasite svoj izbor na ovom papiru.
- Da bi student položio pismeni deo ispita, potrebno je da sakupi barem 17 bodova, od čega barem 1 na Intel-u i barem 1 na Arm-u.

Deklaracija akademskog integriteta

Potpisivanjem ovog dela, garantujem svojom čašću da su odgovori na ovom kolokvijumu moj rad bez pomoći drugih osoba i upotrebe nedozvoljenih materijala.

Potpis studenta:

- (ARM): Dat je string koji se sastoji od brojeva u heksadekadnom zapisu, donje crte (–) koja služi kao separator i otvorenih i zatvorenih zagrada. Garantovano je da svaka otvorena zagrada ima svoju zatvorenu, kao i da svaka zatvorena ima svoju otvorenu. Takodje, zagrade su disjunktne, tj. ne može da se dogodi da se jedan par zagrada nalazi unutar drugog para. Potpis funkcije je **void ObradiString(char *str, char* cifra, int* br_pojavljivanja, int * brSum)**. U glavnom programu, main.c učitava se odgovarajući string, a funkcija koja ga obradjuje treba da uradi dva posla. Van zagrada treba naći najfrekventniju cifru i njen broj pojavljivanja, a unutar zagrada treba izračunati koji je to broj čija je suma cifara najveća. Ukoliko nema cifara van zagrada, staviti bilo koju cifru za najfrekventniju i broj pojavljivanja nula. Ako nema brojeva unutar zagrada, staviti da je traženi broj nula.

ulaz1: ___21c__34(ddd_8a3___1)12__11c_(___)(123)

izlaz1: 1 4 ddd

Objašnjenje: Van zagrada najfrekventnija cifra je 1 i ona se javlja 4 puta. Unutar zagrada su brojevi: ddd sa sumom cifara 36, 8a3 sa sumom cifara 31, 1 sa sumom cifara 1 i 123 sa sumom cifara 6. Najveću sumu cifara ima broj ddd.

- (INTEL): Lars Nyland i Mark Snyder napisali su rad koji se zove **Brze trigonometrijske funkcije korišćenjem Intelovih SSE2 instrukcija** u kome navode procenite ubrzanja računanja prostih trigonometrijskih funkcija ako se one kodiraju u asembleru. Od tri razvoja kosinusa koje su spomenuli, dokazali su da najbrže konvergira Tejlorov razvoj:

$$\cos(x) = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \frac{x^8}{8!} - \dots$$

U main.c učitava se tačka u kojoj se računa ugao u radijanima, kao i broj članova koliko ih treba izračunati. Može se pretpostaviti da važi $-\pi/2 < x < \pi/2$. Po uzoru na njihova razmatranja, zadatak je napisati funkciju koja računa kosinus po ovoj formuli, ali u našem slučaju koristeći paralelne SSE instrukcije. Potpis funkcije je **void Kosinus(float x, int n, float* rezultat)**.

Srećan rad!
