

Napomene:

1. Obavezno pročitati **SVE** napomene.
2. Zadatak snimiti pod imenom **zad.S**. Ovo je jedini fajl koji će biti pregledan.
3. Obavezno upisati **ime, prezime i broj indeksa** u komentar na početku fajla. Nepotpisani zadaci neće biti pregledani.
4. Rešenje koje ne može da se kompajlira nosi **0** poena.
5. Obavezno iskomentarisati kod.
6. Koristiti praktikum iz ispitnih materijala (**ne** kopirati ga kod sebe u direktorijum).
7. Za vreme testa nema pauze. Izlazak iz učionice se smatra završetkom testa.
8. Asistenti za vreme testa pružaju informacije vezane isključivo za razumevanje teksta zadatka i pomoć oko korišćenja programskih alata.
9. Nazivi promenljivih koje se inicijalno nalaze u zad.S fajlu se **ne smeju menjati**. Njihove vrednosti se mogu (i trebaju) menjati, da se ispituju razni ulazi za program. Sadržaji stringova koji su zadati u zad.S fajlu se takođe **ne smeju menjati**. Ukoliko je potrebno, mogu se dodavati nove promenljive.
10. **Test se snima.**
11. Test traje 2 sata i 45 minuta.

Zadatak:

Napisati asemblerski program koji proverava da li je logička adresa validna (da li se nalazi u opsegu $[0, gr-1]$, gde je gr vrednost graničnog registra). Ulaz programa su dva stringa maksimalne dužine 100 znakova, u prvom se nalaze sadržaji registara $r1$ i $r2$ i vrednost graničnog registra gr , dok drugi predstavlja adresiranje memorije. Prvi string se unosi u obliku tri neoznačena 32-bitna dekadna broja razdvojena razmacima (smatrati da će ulaz za prvi string uvek sadržavati tri grupe znakova razdvojene sa tačno jednim razmakom, bez razmaka na početku i kraju). Drugi string se unosi kao adresiranje memorije u obliku

(reg1,reg2,mnozilac)

gde su validne vrednosti za **reg1** i **reg2** jedan od registara iz skupa $\{ "r1", "r2" \}$, a za **mnozilac** je vrednost iz skupa $\{ 1, 2, 4, 8 \}$. Smatrati da će se u ulaznom stringu na mestima **reg1** i **reg2** nalaziti podstringovi dužine 2 znaka, da će se na mestu **mnozilac** nalaziti podstring dužine 1 znak, kao i da će neophodne zagrade i zarezi uvek biti prisutni. Adresa u memoriji se dobija kao **reg1+reg2*mnozilac**. Ako se desi greška prilikom rada programa, ispisati poruku o grešci i završiti rad. Ukoliko se dobije nevalidna adresa, ispisati poruku **greska1**. Ukoliko se desi greška kod unosa vrednosti registara, ispisati poruku **greska2**. Ukoliko se desi greška kod unosa adresiranja, ispisati poruku **greska3**. Ako greške nije bilo, ispisati dekadnu vrednost adrese.

Izlazni kod programa treba da bude **0** ako nije bilo greške, a **1** ako jeste.

Primer upotrebe:

Unesite vrednosti $r1\ r2\ gr$: 123 456 1948

Unesite adresiranje: (r1,r2,4)

Adresa: 1947

Bodovanje zadatka će u velikoj meri zavisiti od procenta uspešnih testova.

Napomena: ukoliko testovi koji ne treba da izazovu grešku ne prolaze, testovi koji treba da izazovu grešku se ne uzimaju kao validni.

Pored testova koji su unapred dati (automatizovano testiranje sa **./testiraj.sh zad.S**), prilikom pregledanja rešenje će se testirati sa još dodatnih testova, te je potrebno testirati i sa drugim ulazima.

Napomena: **testiraj.sh** ima smisla pokretati tek kada je zadatak završen.

Za kompletno odrađen zadatak se dobija 30 poena.