

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 ispitaju 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 za interpretaciju asemblerskih naredbi. Asemblerska naredba je zadata u formatu:

memonik izvorišni_operand,odredišni_operand (npr. ADD \$0x123,%R1)

Izvorišni operand je uvek 32-bitni neoznačen ceo broj zapisan u heksadecimalnom brojnem sistemu kojem prethodi sekvenca "\$0x". Nakon karaktera koji pripadaju izvorišnom operandu uvek sledi zarez. Podrazumevati da će "\$0x" i " " uvek biti ispravno navedeni i to ne treba proveravati.

Odredišni operand je uvek hipotetički registar R1. Registar je kapaciteta 32 bita. Inicijalna vrednost registra na početku rada programa je uvek 0.

Mnemonik je string od tačno tri velika slova iza kojih sledi razmak. Mnemonik će uvek biti jedan od navedena tri:

- ADD – dodaje vrednost izvorišnog operanda na vrednost u registru R1,
- SUB – oduzima vrednost izvorišnog operanda od vrednosti registra R1,
- MUL – sadržaj registra R1 množi izvorišnim operandom.

Mnemonik će uvek biti zadat velikim slovima, nikad neće biti različit od ADD, SUB i MUL i biće praćen tačno jednim razmakom.

Tok programa:

Na početku rada uneti string maksimalne dužine 5 karaktera koji predstavlja broj asemblerskih naredbi. Broj naredbi se zadaje kao 32-bitni neoznačen ceo broj u dekadnom brojnem sistemu. Maksimalan broj naredbi koje se mogu izvršiti je 5. Nakon što se unese broj naredbi, unosi se jedna po jedna asemblerska naredba u gore navedenom formatu. Stringovi koji predstavljaju naredbe su maksimalne dužine 50 karaktera. Pre nego se unese naredna naredba, trenutno unesena naredba se izvršava tako što se nad vrednošću u registru R1 izvrši operacija zadata mnemonikom. Ukoliko se pri izvršavanju naredbe desi neka od grešaka, ne nastavlja se sa daljim unosom naredbi i program se završava ispisom stringa **poruka3**. Ukoliko se pri izvršavanju programa ne desi ni jedna greška, na kraju programa ispisati string **poruka4** i vrednost registra R1. **Ako nema greške, izlazni kod programa treba da bude 0**. Za detalje oko ispisa poruka videti primer izvršavanja programa.

Greške koje treba detektovati:

- neispravan karakter pri konverziji iz znakovnog u interni oblik – **izlazni kod programa je 1**
- prazan string pri konverziji iz znakovnog u interni oblik – **izlazni kod programa je 1**
- prekoračenje pri konverziji iz znakovnog u interni oblik – **izlazni kod programa je 2**
- prekoračenje pri izvršavanju naredbi ADD, SUB i MUL (rezultat naredbe ne može da stane u 32 bita) – **izlazni kod programa je 3**
- uneti broj naredbi je veći od 5 – **izlazni kod programa je 4**

Primer interakcije:

Unesite broj naredbi: 4

Unesite naredbe (naredba po liniji):

ADD \$0x123,%R1

MUL \$0x2,%R1

SUB \$0x245,%R1

ADD \$0x1,%R1

Vrednost u registru R1 nakon operacija: 2

Bodovanje zadatka će u velikoj meri zavistiti 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.