

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 koji od korisnika traži da unese niz 32-bitnih neoznačenih brojeva u dekadnom brojnem sistemu, a zatim taj niz brojeva ispiše u rastućem sortiranom redosledu u oktalnom brojnem sistemu. Elementi niza se unose kao jedan string sa do 150 znakova, razdvojeni sa jednim ili više razmaka. Smatrati da neće biti uneto više od 10 elemenata i da neće biti razmaka na početku i kraju stringa.

Ukoliko se desi greška, treba prekinuti rad programa i ispisati poruku. Ukoliko se unese nekorektan element niza, treba ispisati grešku **greska1** praćenu tim elementom. Ukoliko se unese prazan string, treba ispisati grešku **greska2**. Izlazni kod programa treba da bude **0** ako greške nije bilo, a **1** ako jeste.

Primer upotrebe:

Unseite do 10 dekadnih brojeva: 567 123 5 888 777
Sortirani oktalni niz: 5 173 1067 1411 1570

Napomena: sortiranje se može odraditi bilo kojim algoritmom, jedan od njih je i bubble sort koji u svojoj najjednostavnijoj varijanti izgleda ovako:

```
void bubbleSort(int * niz, int nelem) {
    char zamena = 1;
    while(zamena) {
        zamena = 0;
        for (i = 1; i < nelem; i++) {
            if (niz[i-1] > niz[i]) {
                << zameni elemente niz[i-1] i niz[i] >>
                zamena = 1;
            }
        }
    }
}
```

Bodovanje zadatka će u velikoj meri zavisići 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.5`), 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 komplećno odrađen zadatak se dobija 30 poena.