

Zadaci sa rešenjima dostupnim na sajtu

1. U asemblerskom jeziku napisati program koji izbacuje razmake sa početka stringa:
 - a. Proširiti kod za izbacivanje razmaka sa kraja stringa sa prezentacije za 4. nedelju vežbi
 - b. Zadržati kod za brisanje razmaka sa kraja stringa; konačan string treba da bude takav da ne sadrži razmake ni na početku, ni na kraju stringa
 - c. Kao pomoć, koristiti C kod sa prezentacije za 4. nedelju vežbi
 - d. Ideja – pomerati ceo string za jedno mesto ulevo, sve dok se na početku ne nađe karakter različit od blanko karaktera
 - e. Rešenje – fajl *trim.S*

Tekstovi za ostala tri zadatka nalaze se u odgovarajućim direktorijumima (z1, z2, z3), u fajlovima sa nazivom zadatak.txt. Fajlovi u sva tri direktorijuma su uređeni na način na koji će biti uređeni i fajlovi koji će vam biti dostupni na kolokvijumu (testu), uz tu razliku što direktorijumi sa vežbi sadrže i rešenje:

1. Fajl zadatak.txt sadrži tekst zadatka, kao i primer toga kako treba da izgleda interakcija korisnika sa programom.
2. Fajl testiraj.sh sadrži skriptu koja služi za testiranje programa. Skriptu pokrećete tako što se pozicionirate u direktorijum u kome se nalazi ta skripta, kao i vaše rešenje (.S fajl), pa nakon toga u terminalu kućate `./testiraj.sh moje_resenje.S`. Dakle, skripti prosleđujete fajl sa kodom (.S) fajl, a ne fajl koji ste dobili kao rezultat kompajliranja. Skripta će za vas kompajlirati program i izvršiti ga sa određenim test primerima. Postoji šansa da nećete imati u startu privilegije za pokretanje ove skripte. U tom slučaju, promenite privilegije na jedan od sledećih načina:
 - a. Desni klik na *testiraj.sh* -> *properties* -> otvorite jezičak *Permissions* -> čekirajte polje *Allow executing file as program*
 - b. Otvorite terminal u direktorijumu gde se nalazi *testiraj.sh* skripta -> kućajte:
`chmod +x testiraj.sh`

Nakon pokretanja testa na pomenuti način dobićete za nekoliko test primera informaciju o tome koji ulaz je prosleđen vašem programu, koji je očekivani izlaz, kao i koji izlaz je dao vaš program. Na kraju će biti ispisano i koji broj testova je prošao uspešno.

Napomena: Test čija je uspešnost 100%, odnosno koji prođe sve test primere, nije garancija da su vaš program i vaše rešenje potpuno ispravni. Sami treba da vodite računa o tome da program pokriva granične slučajeve (*edge cases*). Preporuka jeste pisati prvo program koji ćete sami testirati, bez upotrebe skripte, korišćenjem sopstvenih primera i dibagera, pa tek nakon toga pokretati automatsko testiranje pomoću skripte.

3. Fajl sa pripremljenim promenljivama i stringovima, u kome treba pisati rešenje. Ovaj fajl ima različit naziv u svakom od zadataka. Svaki ovakav fajl, kao i onaj koji će vam biti dostupan na kolokvijumu, unapred sadrži određen broj promenljivih i određen broj stringova. Važno je ne menjati nazive promenljivih, kao ni sadržaj stringova. Ukoliko promenite nešto od pripremljenih stvari, nijedan test primer neće proći uspešno. Te promenljive i stringovi su važni jer će ih skripta za testiranje pri izvršavanju očekivati baš u onom obliku u kakvom se unapred nalaze u fajlu. Možete dodavati linije koda pre, nakon i između linija tih već

postojećih stringova i promenljivih, i to neće uticati na izvršenje, ali nemojte menjati sam njihov sadržaj. Svoje rešenje pišete u ovom fajlu, i njega prosleđujete skripti za testiranje.

U nastavku su date napomene vezane za zadatke, kao i određeni dodaci:

z1:

- Ovaj zadatak možete testirati skriptom tek nakon što uradite deo pod b). Ukoliko ne uradite ovaj deo, testovi neće proći uspešno. Promenljiva imeprez namenjena je za prihvatanje unosa sa tastature, maksimalne dužine 50 karaktera (računajući i enter).

z2:

- Osnovni zadatak treba da proverava potpuno poklapanje karaktera sa jedne i druge strane stringa. Dakle, ako imate string *anavolimilovana*, to će biti palindrom, ali string *Ana* neće biti jer je sa jedne strane veliko 'A', a sa druge malo 'a'. Takođe, ako je string *ana voli milovana*, to neće biti palindrom, jer je četvrti karakter sa leve strane razmak, a četvrti sa desne je 'v'. Test skripta će proveravati ovakvo rešenje. Promenljiva unos treba da prihvati unos sa tastature maksimalne dužine 50 karaktera (računajući i karakter za novi red).
- **Dodatak 1:** Modifikovati prvi zadatak tako da ne pravi razliku između malih i velikih slova (*Ana* će biti palindrom).
- **Dodatak 2:** Zanimariti sve što nije karakter prilikom provere (razmake, specijalne znake, cifre; *ana voli milovana* je palindrom; *anA v# oli MiLo?van a* je palindrom).
- Rešenje *palindrom_resenje.S* je rešenje osnovnog zadatka, bez dodataka.
- Rešenje *palindrom_prosireno_resenje.S* je rešenje za zadatak sa oba dodatka. U ovom rešenju fokus je na samom algoritmu, tako da se ne vrši unos sa tastature i ispis na ekran, već se string na početku nalazi u memoriji. Takođe, izlazni kod se smešta u promenljivu, a ne u povratnu vrednost programa.
- Svoje rešenje pišite u pripremljenom fajlu *palindrom.S*.
- Fajl *palindromi-srp.txt* sadrži spisak palindroma koje možete koristiti za ručno testiranje (obratiti pažnju na to da sadrže razmake).

z3:

- **Dodatak:** Ispisati unete stringove međusobno uređene leksikografski. Prvo ispisati onaj čije je prvo slovo leksikografski najmanje (najbliže početku abecede). Ako dva stringa počinju na isto slovo, urediti ih po istom principu na osnovu drugog karaktera. Ako su i drugi karakteri isti, posmatrati treći i td.
- Rešenje *stringlist_resenje.S* je rešenje osnovnog zadatka, bez dodatka.
- Rešenje *stringlist_resenje_dodatak_sortiranje.S* je rešenje zadatka sa dodatkom.
- Svoje rešenje pišite u pripremljenom fajlu *stringlist.S*. Dodatno definišite potrebne promenljive po uputstvu iz teksta zadatka (fajl *zadatak.txt*).

Zadaci bez rešenja dostupnih na sajtu

1. Napisati asemblerski program za sortiranje niza karaktera po abecedi:
 - a. Sortirati u proizvoljnom redosledu
 - b. Rezervisati memorijski prostor i u njega upisati string proizvoljne veličine; sortirati karaktere ovog niza
 - c. Koristiti proizvoljni tip adresiranja
2. Napisati asemblerski program koji spaja dva stringa (ulazni stringovi su dužine do 50 karaktera, izlazni 100). Interakcija sa programom treba da izgleda ovako:

Unesite prvi string: prvi
Unesite drugi string: drugi
Spojeno: prvidrugi