

Zadaci sa rešenjima dostupnim na sajtu

1. U asemblerском језику написати програм који избације размаке са почетка stringa:
 - a. Proširiti код за избацивање размака са kraja stringa sa prezentacije za 4. nedelju vežbi
 - b. Zadržati код за бришење размака са kraja stringa; konačan string treba да буде такав да не садржи размаке ни на почетку, ни на kraju stringa
 - c. Кao помоћ, користити C код са prezentacije за 4. nedelju vežbi
 - d. Ideja – померати цео string за једно место улево, све док се на почетку не нађе карактер различит од blanko карактера
 - e. Rešenje – fajl *trim.S*

Tekstovi за остала три задатка налазе се у одговарајућим директоријумима (z1, z2, z3), у фајловима са називом *zadatak.txt*. Фајлови у сва три директоријума су уређени на начин на који ће бити уређени и фајлови који ће вам бити доступни на колоквијуму (тесту), уз ту разлику што директоријуми са веžби садрže и решење:

1. Фајл *zadatak.txt* садржи текст задатка, као и пример тога како треба да изгледа интеракција корисника са програмом.
2. Фајл *testiraj.sh* садржи скрипту која служи за тестирање програма. Скрипту покрећете тако што се позиционирате у директоријум у коме се налази та скрипта, као и ваše решење (.S фајл), па након тога у терминалу кувате *./testiraj.sh moje_resenje.S*. Дакле, скрипти прослеђујете фајл са кодом (.S) фајл, а не фајл који сте добили као резултат компајлiranja. Скрипта ће за вас компајлirati програм и изврши га са одређеним тестовима. Постоји љанса да нећете имати у startu privilegije за покretanje ове скрипте. У том случају, промените privilegije на један од следећих начина:
 - a. Desni klik на *testiraj.sh* -> *properties* -> отворите језичак *Permissions* -> чекирајте полje *Allow executing file as program*
 - b. Отворите терминал у директоријуму где се налази *testiraj.sh* скрипта -> куцайте:
chmod +x testiraj.sh

Након покretanja теста на пomenuti начин добићете за неколико теста првома информацију о томе који улаз је прослеђен ваšем програму, који је очекивани излаз, као и који излаз је дао ваš програм. На kraju ће бити исписано и који број тестова је прошао uspešno.

Nапомена: Тест чија је uspešnost 100%, односно који прође све тестове, nije гаранција да су ваš програм и ваše решење потпуно исправни. Сами треба да водите рачуна о томе да програм покрива граничне случајеве (*edge cases*). Препорука јесте писати прво програм који ћете сами тестирати, без употребе скрипте, коришћењем свогствених примера и дебагера, па тек након тога покретати аутоматско тестирање помоћу скрипте.

3. Фајл са припремљеним променљивима и stringovima, у коме треба писати решење. Овај фајл има различит назив у сваком од задатака. Сваки овакав фајл, као и онaj који ће вам бити доступан на колоквијуму, unapred садржи одређен број променљивих и одређен број stringova. Важно је не менјати називе променљивих, као ни садржай stringova. Уколико промените нешто од припремљених ствари, нijedan тест први неће проći uspešno. Те променljive и stringovi су важни јер ће ih скрипта за тестирање при изврšavanju очекivati баš у оному облику у квадратном скоба у којима се unapred налазе у фајлу. Можете 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, makismalne 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:** Zanemariti 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 dodatka.
- 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 sl.
- 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 asemblerски program za sortiranje niza karaktera po abecedi:
 - a. Sortirati u proizvolnjom redosledu
 - b. Rezervisati memorijski prostor i u njega upisati string proizvoljne veličine; sortirati karaktere ovog niza
 - c. Koristiti proizvoljni tip adresiranja
2. Napisati asemblerски 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: prvidragi