

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 assemblerski program koji broji koliko ima elemenata u nizu koji su **n** puta veći od njemu prethodnog elementa i bar za **m** manji od sledeceg. Drugim rečima, da bi se element niza brojao, treba da važi sledeći uslov: $n * \text{prethodni} \leq \text{trenutni} \leq \text{sledeci-m}$. Dobijeni broj ispisati na ekran u heksadecimalnom formatu.

Prvi i poslednji element niza se nikada ne broji, pa za njih ne treba proveravati ispunjenost uslova.

Format unosa je sledeći:

- Unos uvek počinje fiksnim delom „n=“ iza kojeg sledi neoznačeni 32-bitni ceo broj u dekadnom brojnom sistemu praćen znakom zarez „,“. Navedeni broj predstavlja vrednost **n** iz opisa zadatka.
- Sledi fiksni deo „m=“ iza kojeg sledi broj u istom formatu kao i **n**.
- Sledi fiksni deo „niz=“ praćen parom vitičastih zagrada, između kojih se nalazi niz neoznačenih 32-bitnih celih brojeva u dekadnom brojnom sistemu. Iza svakog broja sledi znak zarez „,“.
- Karakteri koji stoje iza zatvorene vitičaste zagrade se ne posmatraju.
- Maksimalna dužina unosa je 100 karaktera. Smatrati da neće biti uneto više od 20 elemenata niza. Smatrati da će neophodne zagrade, fiksni delovi stringa i zarezi uvek biti prisutni i to baš po gore navedenom formatu.

Na primer:

$n=2, m=10, \text{niz}=\{100, 200, 210, 210, 500, 1000\}$ ostatak se ne posmatra

Ukoliko dođe do prekoračenja prilikom provere uslova (kod množenja ili oduzimanja), treba ispisati poruku **greska1**.

Ukoliko dođe do greške prilikom konverzije brojeva iz stringa u interni oblik, treba ispisati poruku **greska2**.

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

Primer interakcije sa programom:

Unesite izraz: $n=2, m=10, niz=\{100, 200, 210, 210, 500, 1000\}$, dodatak
Ukupno brojeva koji ispunjavaju uslov: 0x2

Unesite izraz: $n=2g, m=10, niz=\{100, 200, 210, 210, 500, 1000\}$, dodatak
GRESKA: Problem prilikom konverzije brojeva.

Unesite izraz: $n=4294967296, m=10, niz=\{100, 200, 210, 210, 500, 1000\}$, dodatak
GRESKA: Problem prilikom konverzije brojeva.

Unesite izraz: $n=2, m=10, niz=\{100, 200, 210, 210, 500, 0\}$
GRESKA: Prekoracenje kod provere uslova.

(objašnjenje: pošto su brojevi neoznačeni, kad proveravamo poslednji uslov i uradimo 0-10 , doći će do prekoračenja)

Unesite izraz: $n=2, m=10, niz=\{100, 200, 210, 2147483648, 500, 1000\}$
GRESKA: Prekoracenje kod provere uslova.

(objašnjenje: pošto su brojevi 32-bitni, kad proveravamo poslednji uslov i uradimo $2147483648*2$, doći će do prekoračenja)

Unesite izraz: $n=2, m=10, niz=\{\}$
Ukupno brojeva koji ispunjavaju uslov: 0x0

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.