



Pravila

Rješenja zadaće će biti izlagana na vježbama 16.1.2024. Rok za predaju je 15.1.2024. u 12:00 sati. Sav kod zadaće predajte na GitHub u repozitorij MORAS, u mapu MORAS/zadaca.4.

Zadatak 1 (10 + 15).

- (a) Implementacija parsera za jezik virtualnog stroja prezentirana na vježbama sljedeće linije neće smatrati pogrešnima.

```
push local zZzZz
pop argument h4ck
```

Argumenti svih push i pop naredbi bi trebali biti nenegativni cijeli brojevi. Modificirajte implementaciju parsera tako da ona za pogrešne push i pop naredbe vraća odgovarajuću informaciju o pogrešci.

- (b) Isti parser za jezik virtualnog stroja daje mogućnost skakanja pomoću naredbi "goto" i "ifgoto" na oznake (label) koje nisu definirane unutar programa. Modificirajte implementaciju parsera tako da referenciranje nedefiniranih oznaka smatra pogrešnim i vraća odgovarajuću informaciju o pogrešci.

Zadatak 2 (15 + 15 + 15). Implementirajte sljedeće funkcije u jeziku virtualnog stroja.

- (a) Funkcija *max* prima 3 argumenta a, b, c i vraća najveći po vrijednosti.
- (b) Funkcija *pell* prima broj $n \geq 0$ i vraća n -ti pellov broj P_n . Nije potrebno provjeravati ispravnost inputa. Više o pelovim brojevima možete pročitati na https://en.wikipedia.org/wiki/Pell_number.
- (c) Funkcije *div* i *mod* primaju dva nenegativna cijela broja m i n te vraćaju rezultat i ostatak cjelobrojnog dijeljenja m/n . Algoritam za dijeljenje je sljedeći.

```
div(N, D)
  Q = 0
  R = N
  while R >= D do
    Q = Q + 1
    R = R - D
  return (Q, R)
```



Zadatak 3 (30). U jeziku virtualnog stroja implementirajte funkciju *BS.search* koja prima argumente:

- početak sortirano polja (adresa segmenta *THAT*),
- duljina sortirano polja n ,
- element x ,

te u polju $THAT[0..(n-1)]$ algoritmom binarnog pretraživanja pronalazi indeks elementa x . Ukoliko se x ne nalazi u nizu, vratite -1.

Primjer 1.1. Neka je dan niz $A = [2, 3, 4, 6, 7, 8, 9]$. Tada poziv $BS.search(THAT, 7, 4)$ vraća 2, a $BS.search(THAT, 7, 5)$ vraća -1.

Napomena 1.1. Nije preporučeno implementirati rekurzivni algoritam binarnog pretraživanja zbog ograničenosti veličine stoga. Pripazite, konstantni segment ne sadrži negativne brojeve.