PROGRAMSKI PREVODIOCI

Projekat 1

U programskom jeziku Python napraviti interpreter koji izračunava matematičke izraze i ima mogućnost definisanja i poziva funkcija. Pod matematičkim izrazima podrazumevaju se operacije sabiranja, oduzimanja, množenja i deljenja. Unutar izraza je moguće uključiti i zagrade. Operandi su celi brojevi a poziv funkcije takođe može biti argument operatora. Potrebno je uzeti u obzir i rad sa negativnim brojevima.

Interpreter prihvata izraze u beskonačnoj petlji i nakon svakog unetog izraza ispisuje rezulat. Unošenjem ključne reči 'exit' interpreter prekida sa radom.

```
→ 1 + 2 * (3 + 4)
← 15
→ exit
```

Da bi se definisala funkcija potrebno je uneti ključnu reč 'def'. Nakon toga interpreter očekuje unos zaglavlja funkcije, tj. njenog imena i skupa argumenta. Nazivi funkcija i argumenata prate pravila naziva funkcija i promenljivih u programskim jezicima: to su stringovi koji moraju počinjati slovnim karakterom i ne sadrže specijalne karaktere. Pošto se unese zaglavlje funkcije potrebno je uneti izraz koji predstavlja telo funkcije. Telo funkcije može imati samo jedan izraz, tj. samo jednu liniju. Povratna vrednost funkcije je rezultat izraza tela funkcije i uvek je ceo broj. U izrazu tela funkcije mogu učestvovati sve pomenute operacije, a kao operandi pored celih brojeva učestvuju i argumenti funkcije.

```
→ def
_ def →  f(a, b)
_ def →  a + b + 5
→ f(1, 2)
← 8
→ f(1, 1) + f(2, 2)
← 16
```

Moguće je definisati proizvoljan broj funkcija, a ako se funkcija sa istim imenom definiše više puta validna je samo poslednja definicija. U slučaju da je funkcija nepravilno definisana potrebno je vratiti odgovarajuću poruku. Ako dođe do greške u definisanju zaglavlja, poruka treba biti ispisana odmah:

```
→ def
_ def →  f(a,   )
← zaglavlje funckije nije ispravno
```

Ako se telo funkcije ne navede pravilno, o tome je potrebno izvestiti tek kada se funkcija pozove:

```
→ def
_ def →  f(x)
_ def →  a + b
→ f(5)
← funkcija f nije ispravno definisana
```

Postoji mogućnost pozivanja funkcije unutar funkcije.

→ def
_ def → f(x)
_ def → x+1
→ def
_ def → g(x, z)
_ def → f(x) + f(z)
→ g(1, 1)
← 4

Ovo predstavlja potencijalni problem u kome se funkcije pozivaju beskonačno. Ovakvu situaciju treba detektovati nakon poziva funkcije i obustaviti njeno izvršavanje sa određenim obaveštenjem.

→ def
_ def → f(x)
_ def → g(x)
→ def
_ def → g(x)
_ def → f(x)
→ g(1)
← beskonačno pozivanje

Sledi lista slučajeva o kojima treba voditi računa i ispisati odgovarajuću poruku.

1. Greške pri parsiranju kao što su neočekivani karakter ili neočekivani token
2. Funkcija koja je pozvana ne postoji
3. Argument funkcije koji je iskorišćen u izrazu nije naveden u zaglavlju funkcije
4. Pri pozivu funkcije nije naveden određeni broj argumenata
5. Zaglavlje funkcije nije navedeno pravilno
6. Telo funkcije nije navedeno pravilno

Pri izradi zadatka dozvoljeno je koristiti materijale rađene na času. Deo zadatka koji se odnosi na samo izračunavanje izraza se neće ocenjivati ali mora da postoji. Za potrebe ovog zadatka nije dozvoljeno koristiti biblioteke zadužene za proces parsiranja.

Rok za izradu i termin odbrane je ponedeljak 4.11. u terminu vežbi.