

Konstrukcija kompilatora - Kolokvijum 2017.

1. Programski jezik dopušta definisanje funkcija čiji su argumenti, lokalne promenljive i rezultat isključivo celi brojevi. Definicija funkcije počinje ključnom rečju *function* za kojom (opciono) sledi ime rezultatske promenljive i znak =, zatim ime funkcije i u zagradama lista argumenata funkcije. Telo funkcije se navodi u okviru vitičastih zagrada i sadrži niz naredbi dodele, izraza i/ili poziva ugrađene funkcije *print* koja služi za ispis. Jezik podržava aritmetičke izraze nad celim brojevima kao i uslovni ternarni operator (? :). Izvršavanje programa počinje od funkcije sa imenom *main*. Napisati interpretator za dati jezik.

- (a) (već urađeno) Program sadrži samo funkciju *main*, bez drugih funkcija, i njihovih poziva.

```
function main() {
    a = 5;
    b = 6;
    print (a + b * 2) - a / 2;
    print a - b;
    c = a < b;
    print c;
}
```

- (b) Dodati funkcije bez povratnih vrednosti, i njihove pozive.

```
function ispisi_kvadrat(n) {
    a = n*n;
    print(a);
}
function ispisi_zbir(x, y) {
    print(x + y);
}
function ispisi_nulu() {
    print(0);
}
function main() {
    a = 5;b = 6;
    print(a);
    print(b);
    ispisi_kvadrat(a+1);
    ispisi_zbir(a, b-1);
    ispisi_nulu();
}
```

- (c) Omogućiti opcione povratne vrednosti funkcijama.

```
function c = proizvod(a, b) {
    c = a*b;
}
function main() {
    a = 4;
    b = 5;
    print(proizvod(a, b));
}
```

- (d) Pomoću ternarnog operatora, mogu se definisati i rekurzivne funkcije.

```
function f = faktorijel(n) {
    f = n==0 ? 1 : n * faktorijel(n-1);
}
function main() {
    a = 5;
    print(faktorijel(a));
}
```