

Konstrukcija kompilatora - Jul 2016.

praktični deo

1. Unaprediti interpretator paskaloidnog pseudojezika sa ključnim rečima na srpskom jeziku (rađenog na vežbama) dodavanjem mogućnosti prevođenja na LLVM IR.

- (a) Svaki program počinje ključnom rečju *pocetak* i završava se ključnom rečju *kraj*. Jezik podržava standardnu funkciju *ispisi* koja ispisuje na standardni izlaz date celobrojne vrednosti. Jezik se odlikuje aritmetičkim operatorima za rad sa celim brojevima. Brojeve konstante se unose ili dekadno (npr. 123) ili heksadekadno (npr. 0xAF38). Iskazi jezika su razdvojeni karakterom ;.

```
pocetak                                define i32 @main() {
    ispisi(2/5 + 1);                    entry:
    ispisi(3 + 0xB mod 5)                %calltmp = call i32 @print_int(i32 1)
kraj.                                   %calltmp1 = call i32 @print_int(i32 4)
                                        ret i32 0
                                        }
```

- (b) U programu je moguće dodeliti vrednost celobrojnim promenljivim pri čemu se ne zahteva ranija deklaracija. Identifikatori se navode na isti način kao u pascalu i jezik ne razlikuje velika od malih slova. Naredba dodele se označava sa :=. Npr:

```
pocetak
    i := 3 * (4 + 2);
    ispisi(i)
kraj.
```

- (c) Jezik poseduje i iskaz *ako_je* - *onda* : - *inace* :, sa uobičajenim značenjem. Deo *inace* : je moguće izostaviti.

```
pocetak
    i := 6;
    ako_je i > 10 onda:
        ispisi("Broj je veci od 10")
    inace:
        ispisi("Broj nije veci od 10")
kraj.
```

- (d) Jezik se odlikuje i iskazom *dok_je* - *radi_sledece* : sa uobičajenim značenjem. Npr:

```
pocetak
    i := 1;
    dok_je i < 10 radi_sledece:
        pocetak
            ispisi(i);
            i := i + 1
        kraj
kraj.
```

- (e) Interpretator se poziva navodjenjem

pseudokod ime_datoteke

u komandnoj liniji. Ukoliko se navede

pseudokod --nazubi ime_datoteke

umesto interpretacije potrebno je na standardni izlaz ispisati kod nazubljen u skladu sa primerima iz teksta ovog zadatka. Opcija kojom se umesto interpretacije vrši prevođenje je

pseudokod --prevedi ime_datoteke

Od priloženih test primera generisati *main* funkciju u *.ll* datoteci, od koje se prevođenjem *clang*-om zajedno sa pomoćnim *C*-funkcijama (iz priložene datoteke *c_functions.c*) može dobiti izvršni program.