

Prevodioci i interpretatori - Februar 2004.
praktični deo ispita

1. Neka je dat asemblerski jezik, za fiktivni nula-adresni računar. Asembler podržava sledeće instrukcije

<code>push x</code>	postavlja <i>x</i> na stek
<code>pop x</code>	skida vrednost sa vrha steka i postavlja je u promenjivu <i>x</i> ako je navedena
<code>add</code>	skida dve vrednosti sa vrha steka i njihov zbir postavlja na stek
<code>sub</code>	skida dve vrednosti sa vrha steka i njihovu razliku postavlja na stek
<code>mul</code>	skida dve vrednosti sa vrha steka i njihov proizvod postavlja na stek
<code>div</code>	skida dve vrednosti sa vrha steka i njihov količnik postavlja na stek
<code>jmp labela</code>	bezuslovni skok na labelu <i>labela</i>
<code>jz labela</code>	skok na navedenu labelu ukoliko je na vrhu steka 0
<code>jnz labela</code>	skok na navedenu labelu ukoliko na vrhu steka nije 0
<code>cmp</code>	skida dve vrednosti sa vrha steka i postavlja 1 ako su jednake i 0 ako nisu
<code>cmpLT</code>	skida dve vrednosti sa vrha steka i postavlja 1 ako je prva manja od druge i 0 ako nije
<code>cmpGT</code>	skida dve vrednosti sa vrha steka i postavlja 1 ako je prva veća od druge i 0 ako nije
<code>print</code>	štampa vrednost sa vrha steka
<code>stop</code>	prekida izvršavanje programa

Dopuštene labele su oblika `Lxxx` : gde su *x* cifre.

Prevodioci i interpretatori - Februar 2004.
praktični deo ispita

1. Neka je dat asemblerski jezik, za fiktivni nula-adresni računar. Asembler podržava sledeće instrukcije

<code>push x</code>	postavlja <i>x</i> na stek
<code>pop x</code>	skida vrednost sa vrha steka i postavlja je u promenjivu <i>x</i> ako je navedena
<code>add</code>	skida dve vrednosti sa vrha steka i njihov zbir postavlja na stek
<code>sub</code>	skida dve vrednosti sa vrha steka i njihovu razliku postavlja na stek
<code>mul</code>	skida dve vrednosti sa vrha steka i njihov proizvod postavlja na stek
<code>div</code>	skida dve vrednosti sa vrha steka i njihov količnik postavlja na stek
<code>jmp labela</code>	bezuslovni skok na labelu <i>labela</i>
<code>jz labela</code>	skok na navedenu labelu ukoliko je na vrhu steka 0
<code>jnz labela</code>	skok na navedenu labelu ukoliko na vrhu steka nije 0
<code>cmp</code>	skida dve vrednosti sa vrha steka i postavlja 1 ako su jednake i 0 ako nisu
<code>cmpLT</code>	skida dve vrednosti sa vrha steka i postavlja 1 ako je prva manja od druge i 0 ako nije
<code>cmpGT</code>	skida dve vrednosti sa vrha steka i postavlja 1 ako je prva veća od druge i 0 ako nije
<code>print</code>	štampa vrednost sa vrha steka
<code>stop</code>	prekida izvršavanje programa

Dopuštene labele su oblika `Lxxx` : gde su *x* cifre.

Neka je, dalje, dat mali programski jezik koji podržava samo celobrojni tip podataka i na njemu definisane operatore +, -, *, /, prefiksni i postfiksni operator inkrementiranja (++) i dekrementiranja (--) kao i relacijske operatore <, >, <=, >=, ==. Svi operatori imaju isto značenje kao i u jeziku C. Od kontrolnih struktura jezik podržava uslovno grananje (if-then i if-then-else) tipa kao i while petlju pri čemu je sintaksa ista kao u jeziku C. Definisana je još i ugrađena funkcija **print** sa jednim celobrojnim argumentom koja služi sa štampanje.

Napraviti program koji vrši prevodjenje sa datog programskog jezika na dati assembler. Npr.

```

                                push 0
                                pop x
L000:
                                push x
                                push 3
                                compLT
                                jz L001
                                push x
                                push 1
                                add
                                pop x
                                push x
                                print
                                jmp L000
L001:
                                stop

```

Neka je, dalje, dat mali programski jezik koji podržava samo celobrojni tip podataka i na njemu definisane operatore +, -, *, /, prefiksni i postfiksni operator inkrementiranja (++) i dekrementiranja (--) kao i relacijske operatore <, >, <=, >=, ==. Svi operatori imaju isto značenje kao i u jeziku C. Od kontrolnih struktura jezik podržava uslovno grananje (if-then i if-then-else) tipa kao i while petlju pri čemu je sintaksa ista kao u jeziku C. Definisana je još i ugrađena funkcija **print** sa jednim celobrojnim argumentom koja služi sa štampanje.

Napraviti program koji vrši prevodjenje sa datog programskog jezika na dati assembler. Npr.

```

                                push 0
                                pop x
L000:
                                push x
                                push 3
                                compLT
                                jz L001
                                push x
                                push 1
                                add
                                pop x
                                push x
                                print
                                jmp L000
L001:
                                stop

```