

## PROGRAMSKI PREVODIOCI

### DODATNI ZADACI ZA VJEŽBU – GRUPA I

1. Aleksandar je uzeo da istražuje rimsku istoriju, i potreban mu je kalkulator rimskih brojeva. Naime, on naleti na ulaz gdje se nalazi nekoliko rimskih brojeva, i potrebno mu je da sve njih prikaže u dekadnom formatu, arapskim ciframa. Separatori su blanko prostori, novi redovi. Napisati parser koji kao ulaz ima proizvoljan broj rimskih brojeva, i prevodi ih u dekadni format, arapskih cifara. Takođe, na samom kraju, program štampa i koliko je rimskih brojeva konvertovano.

ULAZ	IZLAZ
I	1
II IV	2
LD CV	4
	450
	105
	Broj konverzija:5

2. JSONGrammar je jednostavnija verzija JSON formata, i podržava dvije vrste naredbi: deklarativnu naredbu, i naredbu prikaza. Deklarativna naredba je uvijek u sljedećem formatu: T\_ID = jsonobj, gdje je T\_ID token za promjenljivu, a jsonobj predstavlja objekat u json formatu. Jedan primjer ove naredbe je:

```
x = {
    a: 1,
    b: 1
};
```

Atributi u json objektu su uvijek mala ASCII slova, a vrijednosti su uvijek cijeli brojevi. Primjer naredbe prikaza je sljedeći:

```
x.a;
```

U ovoj naredbi, prvo ide token za identifikator, pa tačka, pa naziv atributa. Ova naredba može da da neki od sljedeća tri rezultata:

- Ako ta promjenljiva ne postoji štampa se poruka „Promjenljiva nije definisana“
- Ako promjenljiva postoji, ali ne i definisan atribut nad tom promjenljivom štampa „Ne postoji zadati atribut nad tom promjenljivom“
- Ukoliko postoji i promjenljiva, i atribut, štampa vrijednost koja je opisana sa tim atributom

Osmisliti parser i gramatiku koja za ulaz ima program u JSONGrammar formatu, i štampa potreban izlaz.

ULAZ	IZLAZ
x = { a: 1, b: 1 };	1 Atribut c ne postoji nad promjenljivom x.

<pre> z = {}; x.a; x.c; y.a; </pre>	Promjenljiva y nije definisana.
-------------------------------------	---------------------------------

3. Stem and leaf display je način predstavljanja statističkih podataka numeričkog tipa. Pretpostavimo da imamo dvocifrene brojeve u određenom nizu: 11, 11, 12, 21, 23, 23, 42. Stem and leaf display se sastoji u sljedećem:
- Prva cifra se proglašava stem vrijednošću
  - Ostatak broja predstavlja leaf vrijednost.
- Tako je za broj 11 stem 1, a leaf 1, a za broj 23 stem 2, leaf 3. Dalje, ovaj display se konstruiše tako da sa lijeve strane se nalaze stem vrijednosti, posložene u sortiranom poretku, od najmanje do najveće, bez preskakanja vrijednosti. Npr. U datom primjeru stem vrijednosti su 1, 2, 3 i 4. Iako ne postoji nijedan primjer koji počinje sa cifrom 3, kompletan segment se navodi. Tako stem and leaf display za zadate primjere izgleda:
- ```

1 | 112
2 | 133
3 |
4 | 2

```
- Osmisliti i kreirati parser koji kao ulaz ima stem and leaf display podataka, a kao rezultat daje sljedeće: koji se broj najviše puta pojavio. Koliko je blokova bez brojeva (kao blok 3 u ovom primjeru), kao i koji blok ima najviše brojeva. Ukoliko ih ima više sa istim brojem, navesti ih sve (u ovom primjeru to su blokovi 1 i 2).

| ULAZ                                              | IZLAZ                                                                                                           |
|---------------------------------------------------|-----------------------------------------------------------------------------------------------------------------|
| <pre> 2   1 3   223 4   5   1234 6   7   4 </pre> | <p>Najčešće se pojavljuje: 32.</p> <p>Broj blokova koji nema brojeva: 2.</p> <p>Blok sa najviše brojeva: 5.</p> |

4. SimpleLogic je jednostavan kalkulator za logičke operacije koji ima elemente programskog jezika. U ovom kalkulatoru postoje dva tipa naredbi – naredba izraza i naredba dodjele. Naredba izraza je ustvari jedan logički izraz koji može biti T (tačan) ili F (netačan). T i F su ključne riječi jezika.
- Naredba dodjele je uvijek oblika T\_VAR = exp, gdje je T\_VAR token za promjenljivu. Promjenljiva se opisuje kao nekoliko velikih ili malih ASCII karaktera.
- Od operacija, SimpleLogic podržava AND, OR i NOT, koji respektivno predstavljaju logičko i, ili i negaciju. Prioriteti operacija su kao u programskom jeziku C ili Java, a moguće je i koristiti zagrade pri definisanju izraza.
- Napisati parser, i osmisliti gramatiku koja kao ulaz ima jedan program napisan u SimpleLogic jeziku (po jednu naredbu u redu), a kao izlaz za svaku naredbu izraza štampa njenu vrijednost.

Ukoliko se u naredbi koristi promjenljiva koja nije definisana, njena podrazumijevana vrijednost je T.

| ULAZ                      | IZLAZ |
|---------------------------|-------|
| x = T                     | T     |
| y = F                     | F     |
| x AND T                   | T     |
| x AND y                   | T     |
| NOT y AND x               | T     |
| NOT (x AND y)             | F     |
| y OR z                    |       |
| (x AND y) OR NOT (y OR z) |       |

5. Alis i Bob svakodnevno razmjenjuju poruke preko društvenih mreža. Hakerskim napadom, jednog dana njihove poruke su se izmiješale. Svaka poruka je zadata sljedećim formatom: HH:MM:SS POŠILJALAC: TEKST PORUKE. Dakle, prvi parametar HH:MM:SS je vrijeme poslate poruke. HH, MM, SS su oznake za sate, minute i sekunde. Pretpostaviti da su sva zadata vremena ispravno unijeta. POŠILJALAC može biti A ili B, nakon čega idu dvije tačke, a nakon njih ide tekst poruke. Tekst se sastoji od slova i znakova interpunkcije. Osmisliti gramatiku i napisati parser koji kao ulazni podatak ima sve poruke konverzacije Alis i Bob, i sortira ih onako kako su se napisale. Takođe, potrebno je nakon sortiranja, ispisati koliko poruka je imala Alis, koliko Bob, i koliko karaktera ima najduža poruka.

| ULAZ                                                                                                                 | IZLAZ                                                                                                                                        |
|----------------------------------------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------|
| 00:10:11 A: Poruka 1<br>23:12:12 B: Poruka 5<br>11:12:13 A: Poruka 2<br>12:21:12 B: Poruka 4<br>11:12:15 A: Poruka 3 | 00:10:11 A: Poruka 1<br>11:12:13 A: Poruka 2<br>11:12:15 A: Poruka 3<br>12:21:12 B: Poruka 4<br>23:12:12 B: Poruka 5<br>Alis: 3, Bob: 2<br>8 |