

# Linguaggi e Traduttori

## Esercitazione di laboratorio n° 3

### 1 Esercizio (mini C)

Si scriva, usando JFLEX e CUP, un programma in grado riconoscere la sintassi di un sottoinsieme del linguaggio C (*mini C*). Dato un file in ingresso tale programma deve indicare se il file sia un sorgente corretto.

In particolare le caratteristiche del linguaggio sono le seguenti:

- Non esistono il `main` e le funzioni: l'intero programma sarà perciò inserito in un'unico file di ingresso che rappresenta il `main`.
- Esistono variabili di tipo `int` e `double` e la possibilità di dichiarare vettori ad una dimensione. Le variabili non possono essere inizializzate in fase di dichiarazione (es. un'istruzione del genere `int a=0;` non è contemplata dal linguaggio).
- Gli indici dei vettori possono essere variabili o numeri interi, ma non espressioni complesse (istruzione di assegnamento corretta: `a[2]=3*b[c]-a[3];`; istruzioni di assegnamento non ammesse: `a[2+4]=0;` o `a[c+1]=2;`).
- Si possono eseguire operazioni di assegnamento (esattamente come in C). Il linguaggio permette l'utilizzo di una particolare istruzione di stampa `print(<variabile>);` che permette di stampare il valore rappresentato dalla variabile con nome `<variabile>` (es. `print(a[2]);` stampa il valore di indice 2 del vettore `a`).
- Sono presenti i costrutti `while` e `if`, esattamente con la stessa sintassi del linguaggio C. Si gestiscano correttamente sia la sintassi in cui una lista di istruzioni è presente all'interno di due parentesi graffe, sia il caso in cui nei rami dell'`if` ci sia solo un'unica istruzione, e quindi le parentesi graffe non sono obbligatorie.
- Le espressioni *booleane* all'interno di `while` e di `if` devono prevedere l'utilizzo degli operatori di confronto `"=="`, `"<"`, `"<="`, `">"`, `">="` e gli operatori booleani `"&"` (AND), `"|"` (OR) e `"!"` (NOT). Si faccia particolare attenzione nella gestione corretta delle **precedenze** degli operatori sopra elencati (es. `if (3+2-a[4] < 3-3*a[c]+1 & b==3 | a[2]<=3*b+1)`).

#### 1.1 Esempio file di input

Un esempio di file di ingresso è il seguente:

```
/* Esempio algoritmo di ordinamento Bubble sort */

double x[5];
int i, j;
double swap;
int pos;

/* Inizializzazione vettore */
x[0] = -2.0;
x[1] = -3.0;
x[2] = 3.0;
x[3] = 5.0;
x[4] = 2.5;

/* Bubble sort */
pos = 5;
while(pos > 0){
    i = 0;
    while (i < pos - 1){
        j = i + 1;
        if (x[i] > x[j]){
            swap = x[j];
            x[j] = x[i];
            x[i] = swap;
        }
        i = i + 1;
    }
    pos = pos-1;
}

/* Stampa risultati */
i = 0;
while(i<5){
    print (x[i]);
    i = i + 1;
}
```

## 2 Esercizio

Realizzare mediante Jflex e Cup uno scanner e un parser che riconoscano un linguaggio per la gestione di una biblioteca.

Il file di ingresso è suddiviso in due sezioni separate dal simbolo “%%” (Due simboli percentuale).

La prima sezione è composta da una lista non vuota di scrittori e dei relativi libri da loro scritti. Ogni elemento della lista ha i seguenti campi:

```
<nome scrittore> > <lista di libri>;
```

Dove <nome scrittore> è una stringa di lettere racchiusa tra i caratteri ” (Doppio apice). <lista di libri> è una lista non vuota di libri scritti da un determinato scrittore e separati da una “,” (virgola).

Ogni elemento della lista è così composto:

```
<codice ISBN>:<titolo libro>:<numero di pagine>:<collocazione>
```

<codice ISBN> è formato da due caratteri numerici, seguiti da un trattino, seguito da due caratteri numerici, seguiti da un trattino, seguito da 5 caratteri esadecimale, seguiti da un trattino e seguito da una lettera o da un carattere numerico. <collocazione> (è opzionale) ed è composta dalla parola LI o LS (letteratura straniera o italiana) seguita dal genere AV, BO o SO (Avventuroso, biografico o sociale), seguito da un numero intero ed eventualmente seguito da una lettera. Non esiste però il genere LI BO, lo si gestisca.

La seconda sezione è composta da una lista non vuota di utenti. Ogni elemento della lista è così definito:

```
<nome utente>:<lista di prestiti>;
```

<lista di prestiti> è un insieme di prestiti fatti da un determinato utente della biblioteca, separati dal carattere “,” (virgola).

Per ogni prestito viene riportata una data e il codice ISBN del libro preso in prestito. La data ha il formato “GG/MM/AAAA”, dove GG è un numero tra 01 e 31, MM è un numero tra 01 e 12

Il programma dovrà riconoscere il linguaggio precedente descritto e scrivere se è corretto dal punto di vista grammaticale.

### 2.1 Esempio file di input

```
"Hesse Herman" -> 88-17-83457-X:"Narciso e Boccadoro":200:LS SO 127 A,
                    88-14-24B43-2:"Siddhartha" : 236 : LS SO 127 B,
                    88-12-34AA3-B:"Lupo della steppa, Il":262:LS SO 127 C;
"Baricco Alessandro"-> 88-17-10625-9:"Seta":100:LI AV 1,
                       88-17-86563-X:"City":319:LI AV 2 A;
"F. Christiane"-> 88-17-11520-7:"Noi, i ragazzi dello zoo di Berlino":346:LS BO 1;
%%
"Giovanni": 02/10/2006 88-17-11520-7;
"Stefano" : 12/04/2007 88-17-83457-X,
           20/09/2007 88-14-24B43-2,
           29/09/2007 88-17-11520-7;
"Giovanni": 02/10/2007 88-17-10625-9,
           02/10/2007 88-17-86563-X;
```

## 3 Esercizio (Facoltativo)

Come estensione del Esercizio 1, realizzare la grammatica in grado di riconoscere il seguente sottoinsieme del linguaggio C.

### 3.1 Sottoinsieme del C da riconoscere:

- Dichiarazione di variabili di tutti i tipi predefiniti (con eventuali modificatori `signed` ed `unsigned`), array e puntatori.
- Le definizioni di funzioni con un numero generico di argomenti (da 0 ad  $n$ ) ed un valore di ritorno scelto fra i tipi predefiniti.
- Uso di espressioni aritmetiche o booleane che possano contenere variabili e funzioni in tutte le forme indicate in precedenza.

- Uso di costrutti condizionali `if-else`, `switch`, `while`, `do-while` e `for`.

### 3.2 Sottoinsieme del C da non riconoscere:

- Dichiarazione di tipi mediante `typedef`, dichiarazione ed uso di strutture (`struct`) e di union (`union`), uso di `enum`.
- Variabili che rappresentino puntatori a funzione.
- Cast

### 3.3 Esempio file di input

```
extern int *fn1(int a, int b, char *c[]);
register int ff;

int fn2() {
    static unsigned long int k = 1, i;
    for(i = 0; i < 10; i++) {
        k-1;
    }
}

int main() {
    char *miovett[] = {"Inverno", "Estate"};
    while(fn1(2,3, miovett) != 0) {
        ff++;
    }
    return ff;
}
```