

Linguaggi e Traduttori

Laboratory 6

1 Esercizio (mini C - Gestione errori sintattici)

Beginning with the scanner and the parser for the **mini C** language realised in the past labs, obtain a translator that:

- taking in input a file written in **mini C** syntax
- produces a file written in **Pseudo Assembler** syntax

The syntax of the **Pseudo Assembler** language is reported in the document with the title “Pseudo Assembler interpreter”, that can be downloaded from the site of the course. Also the interpreter that, taking in input a program written in “Pseudo Assembler” executes it, can be downloaded from the site of the course.

For the translation of constructs like **if** and **while**, inherited attributes must be used in order to manage in a correct way the label names.

1.1 Input

```

/* 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] = 2.5/3;

/* Bubble sort */
pos = 5;
while(pos > 0.0){
    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;
}

```

1.2 Output

```

DOUBLE x[5]
INT i
INT j
DOUBLE swap
INT pos
EVAL -2.0
ASS x[0]
EVAL -3.0
ASS x[1]
EVAL 3.0
ASS x[2]
EVAL 5.0
ASS x[3]
EVAL 2.5
ASS x[4]
EVAL 5
ASS pos
L1: EVAL pos 0 > /* while (line 18) */
GOTOF L2
EVAL 0

```

```
ASS i
L3: EVAL i pos 1 - < /* while (line 20) */
GOTO L4
EVAL i 1 +
ASS j
EVAL x[i] x[j] > /* if (line 22) */
GOTO L5
EVAL x[j]
ASS swap
EVAL x[i]
ASS x[j]
EVAL swap
ASS x[i]
L5: EVAL i 1 +

ASS i
GOTO L3
L4: EVAL pos 1 -
ASS pos
GOTO L1
L2: EVAL 0
ASS i
L6: EVAL i 5 < /* while (line 35) */
GOTO L7
PRINT x[i]
EVAL i 1 +
ASS i
GOTO L6
L7: END
```