

Esercizio 1

Si progetti un programma per la traduzione automatica dalla lingua italiana a quella inglese. Questo è basato sulla semplice traduzione parola per parola avendo a disposizione un vocabolario in forma elettronica. Si descriva la struttura dati che utilizzereste per memorizzare il vocabolario bilingua e si descrive almeno lo pseudocodice di tale programma.

Esercizio 2

Si calcoli la complessità computazionale del seguente frammento di codice:

```
1 for (x=1;x<n;x++){
2     y=x*x; c=0;
3     while (y<n){c++;}
4     for (y=0;y<x;y++);
5 }
```

Esercizio 3

Con riferimento al seguente listato:

1. Individuare e correggere gli errori di sintassi che pregiudicano la corretta compilazione del programma.
2. Spiegare a parole lo scopo e il funzionamento del programma.
3. Dire cosa viene stampato richiamando il programma con i parametri ‘‘1 2 3 + /’’.
4. Discutere alcune modifiche da apportare per evitare eventuali malfunzionamenti.

```
1 #include<string.h>
2 #include<stdio.h>
3
4 struct elemento {
5     int *val;
6     struct elemento *succ;
7 };
8
9 typedef struct elemento elemento_t;
10
11 struct struttura {
12     int num;
13     elemento_t *primo;
14 };
15
16 typedef struct struttura struttura_t;
17
18 void elemento_free(elemento_t *p)
19 {
20     free(p.val);
21     free(p);
22 }
23
24 struttura_t struttura_new()
25 {
26     struttura_t result;
27     result.num=0;
28     result.primo=null;
29     return result;
30 }
31
32 void inserisci(struttura_t *str, int num)
33 {
```

```
34     elemento_t *paus;
35     paus=(elemento_t*) malloc (sizeof(elemento_t));
36     paus->val = (int*) malloc (sizeof(int));
37     paus->val=num;
38     paus->succ=str->primo;
39     str->primo=paus;
40     str->num++;
41 }
42
43 int preleva(structura_t* str)
44 {
45     if(str->num) {
46         elemento_t* paus=str->primo;
47         str->primo=str->primo->succ;
48         str->num--;
49         return *(paus->val);
50     }
51
52 int calcola(structura_t* str, char op)
53 {
54     int op1, op2, res;
55     op2=preleva(str);
56     op1=preleva(str);
57     printf(" Calcolo %d%c%d=" ,op1,op,op2);
58     switch('op') {
59     case '+': {
60         res=op1+op2;
61         break; }
62     case '-': {
63         res=op1-op2;
64         break; }
65     case '*': {
66         res=op1*op2;
67         break; }
68     case '/': res=op1/op2;
69     }
70     inserisci(str,res);
71     return res;
72 }
73
74 int main(int argc, char*argv[])
75 {
76     struttura_t strutturale=struttura_new();
77     int i;
78
79     for(i=1;i<argc;i++) {
80         if ((argv[i][0]>="0")&&(argv[i][0]<="9"))
81             inserisci(&strutturale,atoi(argv[i]));
82         else if ((strcmp(argv[i],"+")||strcmp(argv[i],"-")||
83                 strcmp(argv[i],"*")||strcmp(argv[i],"/"))
84                 printf("%d\n",calcola(&strutturale,argv[i][0]));
85     }
86     return 1;
87 }
```

Soluzione Esercizio 3

1. Errori:

- A riga 20 bisogna accedere al capo val con l'operatore corretto, visto che p e' un puntatore: `free(p->val);`
- A riga 28 la parola chiave NULL va scritta in maiuscolo `result.primo=NULL;`
- A riga 37 bisogna deferenziare il puntatore paus per accedere al suo campo val `*paus->val=num;`
- In `preleva()` manca una parentesi graffa chiusa prima del `return`
- A riga 58 `op` e' una variabile quindi non ci vanno gli apici `switch(op)`
- A riga 80 mettere gli apici singoli al posto di quelli doppi, visto che ci riferiamo ad un carattere singolo e non ad una stringa `if ((argv[i][0]>='0')&&(argv[i][0]<='9'))`

2. E' un programma che implementa una calcolatrice in notazione polacca tramite l'utilizzo di uno stack di interi.

3. Richiamando il programma con i parametri `'1 2 3 + /'` viene stampato:

Calcolo 2+3=5

Calcolo 1/5=0

4. Modifiche da apportare per evitare eventuali malfunzionamenti.

- Nella `preleva()` invocare `elemento_free()` (considerabile anche come errore al punto 1)
- Gestire i casi in cui i parametri siano alfanumerici (1A 2b _)
- Controllare l'effettiva dispobilità di operandi nello stack
- Utilizzare float invece di int