

# PROGRAMMAZIONE 1 e LABORATORIO (A,B) - a.a. 2012/13

## Prova scritta del 25 giugno 2013

Scrivere **IN STAMPATELLO** COGNOME, NOME, MATRICOLA e CORSO su ogni foglio consegnato

### ESERCIZIO 1

Si definisca una grammatica libera che genera il linguaggio

$$\mathcal{L} = \{a^n b^m c^k \mid n, m, k > 0\} \setminus \{a^n b^n c^k \mid n, k > 0\}$$

sull'alfabeto  $\Lambda = \{a, b, c\}$ .

### Soluzione

```
S ::= DC
C ::= c | cC
D ::= aDb | aAb | aBb
A ::= a | aA
B ::= b | bB
```

### ESERCIZIO 2

Dato il tipo:

```
type 'a righttopleft = End of 'a | Cont of 'a righttopleft * 'a
```

si definisca una funzione `sum` con tipo

```
sum : int righttopleft -> int
```

in modo che `(sum rtl)` restituisca la somma di tutti gli interi in `rtl`.

### Soluzione

```
let rec sum rtl = match rtl with
  End x -> x |
  Cont (rt, x) -> x + (sum rt);;
```

### ESERCIZIO 3

Scrivere una funzione C che, dati due array di interi e le loro dimensioni, controlla che contengano gli stessi valori, anche se ripetuti un numero diverso di volte.

#### Soluzione

```
boolean check (int a[], int dima, int b[], int dimb)
{
    boolean ok = true;
    int i = 0;
    while (i<dima && ok)
    {
        int j=0;
        boolean trovato=false;
        while (j<dimb && !trovato)
            if (a[i]==b[j]) trovato = true; else j++;
        ok = trovato;
        i = i+1;
    }
    i = 0;
    while (i<dimb && ok)
    {
        int j=0;
        boolean trovato=false;
        while (j<dima && !trovato)
            if (b[i]==a[j]) trovato = true; else j++;
        ok = trovato;
        i = i+1;
    }
    return ok;
}
```

### ESERCIZIO 4

Senza utilizzare ricorsione esplicita, ma solo funzioni di ordine superiore, si definisca in CAML una funzione che, data una lista di interi, restituisce la coppia costituita dal primo elemento maggiore di 0 e dall'ultimo elemento minore di 0, assumendo che la lista contenga almeno un elemento positivo e almeno un elemento negativo.

Ad esempio, foo [-1;0;2;3;-5;-6;0;0] = (2,-6).

#### Soluzione

```
let foo lis = let f x (p,n) = if x>0 then (x,n) else if x<0 and n=0 then (p,x) else (p,n)
              in foldr f (0,0) lis;;
```

## ESERCIZIO 5

Date le seguenti definizioni:

```
struct el {int info; struct el *next;};
typedef struct el ElementoDiLista;
typedef ElementoDiLista *ListaDiElementi;
```

scrivere in C una procedura che, data in ingresso una lista di interi, cancella il primo elemento che è preceduto e seguito da due elementi uguali e lascia la lista inalterata se tale elemento non esiste.

## Soluzione

```
void p (ListaDiElementi lis)
{
    if (lis !=NULL)
        if (lis->next != NULL)
            {
                ListaDiElementi prec=lis, corr=lis->next, succ=corr->next;
                boolean trovato = false;
                while (succ !=NULL && !trovato)
                    {
                        if (prec->info == succ->info)
                            {
                                prec->next=succ;
                                free(corr);
                                trovato=true;
                            }
                        else
                            {
                                prec=corr;
                                corr=succ;
                                succ=succ->next;
                            }
                    }
            }
}
```

### ESERCIZIO 6

Si definisca una procedura C che, dato un array di interi e la sua dimensione, raddoppia tutti gli elementi che sono preceduti nell'array da sé stessi.

Ad esempio, dato l'array

2	8	6	2	10	6	6	4
---	---	---	---	----	---	---	---

la procedura lo deve modificare come segue

2	8	6	4	10	12	12	4
---	---	---	---	----	----	----	---

### Soluzione

```
void fie (int a[], int dim)
{
    int i;
    for (i=dim-1; i>0; i--)
    {
        boolean trovato = false;
        int j=0;
        while (j < i && !trovato)
            if (a[i]==a[j]) { a[i]=2*a[i]; trovato=true; }
            else j=j+1;
    }
}
```