

PROVA SCRITTA
FONDAMENTI DI INFORMATICA E LABORATORIO T-AB
22 DICEMBRE 2008

Esercizio 1 (punti 6)

Si scriva una funzione ricorsiva `float mm(float a, float b, int j);` che calcoli il seguente valore

$$\sum_{i=1}^j (b/a - i/b)$$

Esercizio 2 (punti 10)

Dato il seguente programma C:

```
#include <stdio.h>
#include <stdlib.h>
#define DIM 7

char W[DIM] = {'O', 'L', 'S', 'T', 'I', 'R', 'O'};
int V[DIM] = {12, 11, 8, 4, 3, 2, 1};

int X(int n, int k)
{
    int i;
    V[k] = 0;
    for(i = 0; i < k; i++)
        V[k] += k;
    return V[k]/3 + n;
}

void main()
{
    int i;
    char* W2 = (char*) malloc(sizeof(char)*DIM*2);

    for(i = 0; i < DIM; i++){
        W2[i] = W[i];
        W2[i+DIM] = W[i];
        printf("%c\t", W2[i+DIM]);
    }
    printf("\n");

    for(i = 0; i < DIM; i++){
        int t = X(V[i], i);
        W[i] = W2[t];
        printf("%d\t", t);
    }
    printf("\n");
    for(i = 0; i < DIM; i++){
        printf("%c\t", W[i]);
    }
    free(W2);
}
```

Cosa viene stampato dal programma? La risposta deve essere opportunamente motivata.

Si spieghi come ci si comporta per l'identificatore `i` definito sia nel `main` sia nella funzione.

Cosa fa la `#define` all'inizio del programma.

Esercizio 3 (punti 7)

Data la seguente funzione ricorsiva:

```
int mag(int x, int y){
if (x <= y)
{ if (x == y) return x+2;
else return y-3;}
    else return x - y + mag(x-3,y+2);
}
```

Si dica qual è il valore restituito dalla funzione e si disegnino i record di attivazione nel caso in cui la funzione sia chiamata con i seguenti parametri attuali: **mag(20, 7)**. Si dica se la funzione è tail ricorsiva e si motivi la risposta.

Esercizio 4 (punti 7)

Cosa è un puntatore. Si facciano alcuni esempi di uso degli operatori di indirizzamento (&) e dereferenziazione. (*).

Cosa è un file binario e che tipo di dati contiene?

Cosa è il ciclo fetch/decode/execute chi lo esegue e a cosa serve.

Cosa si intende per algoritmo e qual è il procedimento per passare dalla specifica di un problema al programma che lo risolve.

Cosa è il tempo di vita di una variabile e il suo campo di visibilità? Dove sono visibili le variabili definite all'interno di un blocco? E cosa succede al tempo di vita di una variabile se è allocata dinamicamente tramite malloc sempre all'interno di un blocco?

SOLUZIONE

Esercizio 1

```
float mm(float a, float b, int j)
{if (j == 1) return b/a - j/b;
  else return b/a - j/b + mm(a,b,j-1);
}
```

Esercizio 2

O	L	S	T	I	R	O
12	11	9	7	8	10	13
R	I	S	O	L	T	O

Esercizio 4 (punti 8)

Il risultato ottenuto è 34

