

**PROVA SCRITTA DI FONDAMENTI DI INFORMATICA L-A**  
**Informatici (matricole). Durata ore 2**  
**Docente: Michela Milano - 9 luglio 2002**

**Esercizio 1 (punti 8)**

Si scriva una funzione ricorsiva `int s(int n)`; che calcoli il seguente valore

$\sum_{i=1}^n \text{pow}(2,i)/i$ , essendo `pow(a,b)` la funzione potenza che si suppone già disponibile.

**Esercizio 2 (punti 8)**

Dato il seguente programma C:

```
#include <stdio.h>

#define N 6

void boh(int Q[], int R[]);
void stampa(int vet[],int s, int n);

main()
{
    int uno[N]={1,1,2,3,4,7};
    int due[N]={10,20,30,40,50,60};
    int s = 2;
    stampa(uno,s,N);
    boh(uno,due);
    s++;
    stampa(uno,s-1,N);
    stampa(due, s-1,N);
    printf("%d ",s);}

void boh(int Q[], int R[])
{int i;
  for(i=0;i<N;i++)
    if (Q[i]*10 < R[i])
      Q[i] = R[i];
    else Q[i] = 70;}

void stampa(int vet[], int s, int n)
{int i;
  for(i=0;i*s<n ;i++)
    printf("%d ",vet[s*i]);
  printf("\n");}
```

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

### Esercizio 3 (punti 5)

E' dato un file di testo PARTITE nel quale sono memorizzate strutture (*struct partita*) del tipo:

- codice identificativo prima squadra (intero)
- codice identificativo seconda squadra (intero)
- cognome arbitro (stringa di 20 caratteri, al piu', senza spazi intermedi)

Si scriva un programma C che:

1. chieda all'utente il codice identificativo di una squadra;
2. stampi a video i cognomi degli arbitri che hanno arbitrato le partite giocate dalla squadra.

### Esercizio 4 (punti 7)

Data la seguente funzione ricorsiva:

```
double pot(double x, double y, double z){
    if (z==1) return x * pow(y,x);
    else return pot (x,y+1,z-1) + x + z;}

```

Si dica se la funzione è tail ricorsiva motivando la risposta.

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 **pot(3,1,3)**.

### Esercizio 5 (punti 2)

Si considerino le seguenti espressioni IF e si segni la risposta corretta motivandola opportunamente:

Espressione 1:            **if (a < b);**  
                             **else count++;**

Espressione 2:            **if (a >= b) count++;**

- A) Le due espressioni sono corrette ed equivalenti
- B) L'espressione 1 e' sintatticamente scorretta
- C) Le due espressioni sono corrette ma non sono equivalenti

## Soluzioni

### Esercizio 1

```
#include <math.h>

int s(int n)
{ if (n==1) return 2;
  else return (pow(2,n)/n + s(n-1));
}
```

### Esercizio 2

La procedura `stampa()` stampa a video 1 2 4, cioè gli elementi di indice pari del vettore `uno`. Poi viene invocata la procedura `boh()` che modifica il vettore `Q` in funzione dei valori di `R` in modo che, al ritorno al chiamante, il vettore `uno` risulti: `uno[]={70,20,30,40,50,70}`. La variabile `s` viene poi incrementata di uno; la procedura `stampa()` viene invocata con il secondo parametro con valore `s-1`, cioè 2 e produce 70 30 50 (elementi di indice pari del vettore `uno`) e la successiva chiamata stampa a video il vettore `due[]`: 10 30 50. Al termine del programma, viene poi mostrato il valore di `s`, 3.

Riassumendo, a video compare:

```
1 2 4
70 30 50
10 30 50
3
```

### Esercizio 3

```
#include <stdio.h>
#include <stdlib.h>

struct partita{
    int squadra1;
    int squadra2;
    char arbitro[21];};

main(){
    int team;
    struct partita part;
    FILE *f1;

    printf("\nInserire il codice di una squadra: ");
    scanf("%d",&team);
    if ((f1=fopen("PARTITE.TXT", "r"))==NULL) {
        printf("Il file non esiste!"); exit(1); }
}
```

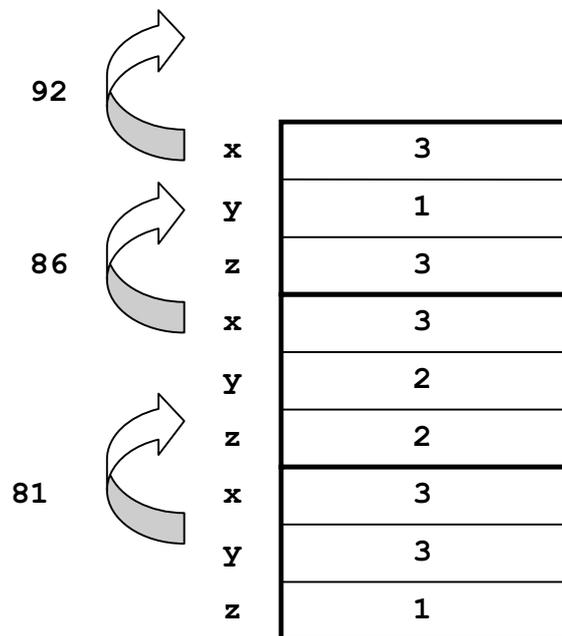
```

while(fscanf(f1,"%d%d%s", &part.squadra1, &part.squadra2, part.arbitro)
!= EOF)
    if ((part.squadra1 == team) || (part.squadra2 == team)) {
        printf("\nArbitro: %s", part.arbitro);
    }
}

```

**Esercizio 4**

La funzione non è tail ricorsiva.



**Esercizio 5**

Le due espressioni sono corrette ed equivalenti.