

PROVA SCRITTA DI FONDAMENTI DI INFORMATICA L-A
Informatici (matricole). Durata ore 2,30
Docente: Paola Mello - 9 luglio 2002

Esercizio 1 (punti 6)

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 6)

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 9)

Sono dati due file testo ARBITRI e PARTITE.

Nel primo file ARBITRI sono memorizzate strutture (struct arbitro) del tipo:

- codice partita (intero)
- cognome arbitro (stringa di lunghezza massima 20, senza spazi intermedi)

Nel secondo file PARTITE sono memorizzate strutture (struct partita) del tipo:

- codice identificativo prima squadra (intero)
- codice identificativo seconda squadra (intero)
- codice partita (intero)

Si scriva un programma C che:

1. chieda all'utente il codice identificativo di una squadra;
2. per ogni elemento del file PARTITE contenente team (come prima o come seconda squadra) cerchi nel file arbitri l'elemento corrispondente al codice partita e stampi a video il numero di partite in cui e' coinvolta la squadra e i cognomi degli arbitri corrispondenti.

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 disegni 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

Esercizio 6 (punti 2)

Si consideri la grammatica G con scopo S e simboli terminali {x,y,z,+,-,(,)}.

$S ::= (C) B (C) \mid C$

$C ::= A B A \mid A$

$B ::= + \mid -$

$A ::= x \mid y \mid z$

Si dica se la stringa $(x+y) - (z)$ e' sintatticamente corretta rispetto a tale grammatica e se ne mostri la derivazione *left most*.

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 arbitro{
    int codpar;
    char carbitro[20];
};

struct partita{
    int squadra1;
    int squadra2;
    int codpar;
};

main()
{
    int i=0,team;
    struct arbitro arb;
    struct partita part;
```

```

FILE *f1, *f2;

printf("\nInserire il codice di una squadra: ");
scanf("%d",&team);

if ((f1=fopen("PARTITE.TXT", "r"))==NULL) {
    printf("Il file non esiste!"); exit(1); }
if ((f2=fopen("ARBITRI.TXT", "r"))==NULL) {
    printf("Il file non esiste!"); exit(1); }

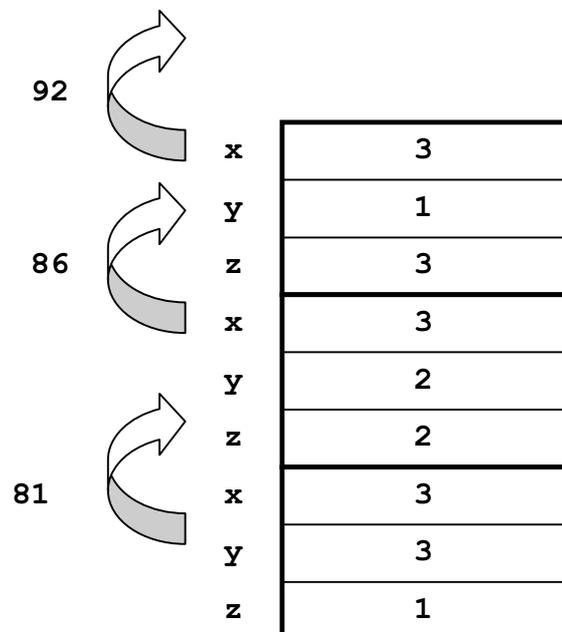
while(fscanf(f1,"%d%d%d", &part.squadra1, &part.squadra2, &part.codpar)
!= EOF)
    if ((part.squadra1 == team) || (part.squadra2 == team)) {
        i++; rewind(f2);
        while(fscanf(f2,"%d%s", &arb.codpar, arb.carbitro) != EOF)
            if (arb.codpar == part.codpar) printf("\nArbitro: %s", arb.carbitro);
        }

printf("\nnumero partite %d\n", i);
}

```

Esercizio 4

La funzione non è tail ricorsiva.



Esercizio 5

Le due espressioni sono corrette ed equivalenti.

Esercizio 6

La stringa e' sintatticamente corretta.

Derivazione *left most*:

S \rightarrow

(C)B(C) \rightarrow (A B A)B(C) \rightarrow (x B A)B(C) \rightarrow (x + A)B(C)
 \rightarrow (x + y)B(C) \rightarrow (x + y) - (C) \rightarrow (x + y) - (A) \rightarrow (x + y) - (z)