

PROVA SCRITTA DI FONDAMENTI DI INFORMATICA A
Prof. Mello, Prof. Milano
12 LUGLIO 2001

Esercizio 1 (punti 9)

Si scriva una funzione ricorsiva `double f(double a, int n);` che calcoli il seguente valore

$$\sum_{i=1}^n (a - i \backslash a)$$

Esercizio 2 (punti 8)

Dato il seguente programma C:

```
#include<stdio.h>
#define Dim 5
char c= 'a';
char A[Dim]={'c','i','a','o','\0'};
char vet[Dim]={'e','e','e','e','\0'};

void sub(char vet1[], char vet2[]);
void stampa(char vet[]);

main()
{stampa(A);
 sub(A,vet);
 stampa(vet);
 stampa(A);
 printf("%c",c);}

void sub(char vet1[Dim], char vet2[Dim])
{int i; char c='b';
 for(i=0;i<Dim-1;i++)
  if (vet2[i]>vet1[i])
   vet1[i]=vet2[i];
  else vet1[i]=c;
 }

void stampa(char vet[])
{printf("Vettore:\n");
 printf("%s ",vet);
 printf("\n");
 }
```

Cosa viene stampato dal programma? La risposta deve essere opportunamente motivata. Si dica inoltre se la variabile `i` definita nella procedura `sub` è visibile dalla procedura `stampa` e dal `main`.

Esercizio 3 (punti 4)

Dato un file di testo **mesi.txt**, si supponga che contenga righe ciascuna contenente una stringa (nome del mese) ed un intero (numero di giorni). Ad esempio:

gennaio	31
febbraio	28
marzo	30

Si stampino a video i nomi dei mesi che hanno 31 giorni.

Esercizio 4 (punti 7)

Data la seguente funzione ricorsiva:

```
double somma_potenza(double a, double b, double n)
{ if (n==1) return a * pow(b,a);
  else return a + n + somma_potenza(a,b,n-1);
}
```

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 **somma_potenza(3,2,3)**.

Esercizio 5 (punti 2)

Qual è la differenza tra un processo computazionale ricorsivo e uno ricorsivo tail ?

- A. Il primo è molto più efficiente del secondo perché nel secondo caso vengono allocati tanti record di attivazione quante sono le chiamate, nel primo uno solo.
- B. Il secondo è molto più efficiente del primo perché nel secondo caso si può effettuare una ottimizzazione che limita il numero di record di attivazione allocati sullo stack.
- C. Nessuna differenza

SOLUZIONE

Esercizio 1

```
double f(double a, int n)
{ if (n==1) return a - 1/a;
  else return a - n/a + f(a,n-1);
}
```

Esercizio 2

La prima stampa produce i valori `ciao` che corrispondono al vettore **A** inalterato. Dopo di che viene chiamata la procedura **sub** che modifica i valori di **vet** e di **A**. Se l'elemento di **vet** è maggiore alfabeticamente del corrispondente elemento di **A**, quest'ultimo elemento viene sovrascritto dall'elemento di **vet**, altrimenti viene sovrascritto con il carattere 'b'.

Quindi **vet** diventa `ebeb` e questo vettore modificato. Viene poi stampato il vettore **A**, producendo `eeee`. Poi viene stampato `vet ebeb`.

Infine viene stampato `c`, che è la variabile definita esternamente al main e non quella definita nella procedura **sub**.

Quindi, il risultato stampato è

```
Vettore:
ciao
Vettore:
eeee
Vettore:
ebeb
a
```

la variabile `i` definita nella procedura **sub** non è visibile nella procedura **stampa** e nel main.

Esercizio 3

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

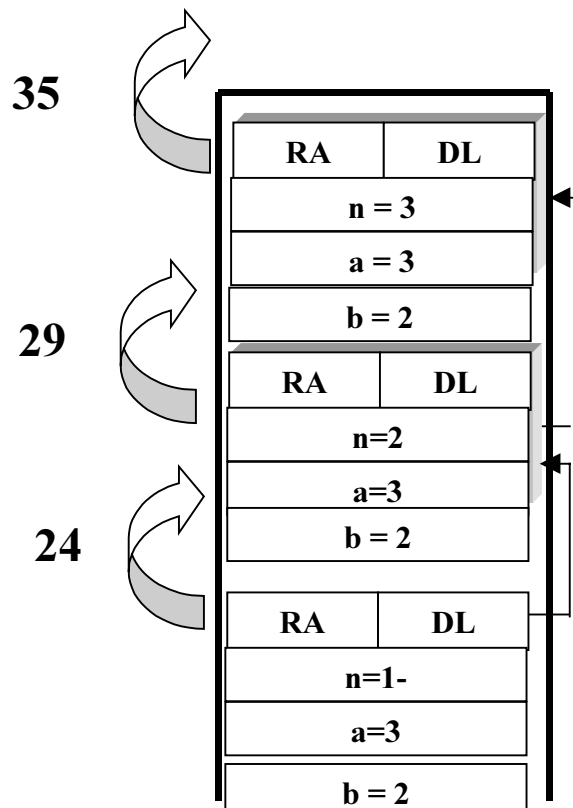
main() {
int giorni; FILE* f;
char nome[20];
if ((f=fopen("mesi.txt", "r"))==NULL) {
    printf("Il file non esiste!"); exit(1); }
while(fscanf(f,"%s %d\n", nome, &giorni) != EOF)
    if (giorni == 31)
        printf("%s\n", nome);
fclose(f);
}
```

Esercizio 4

La funzione non è tail ricorsiva, perché dopo la chiamata ricorsiva deve ancora essere calcolato il prodotto.

Sequenza chiamate

$f(3, 2, 3) \rightarrow f(3, 2, 2) \rightarrow f(3, 2, 1)$



Esercizio 5 (punti 2)

La risposta corretta è la B.