

PROVA SCRITTA DI FONDAMENTI DI INFORMATICA L-A
6 DICEMBRE 2006
COMPITO A

Esercizio 1 (punti 6)

Si scriva una funzione ricorsiva `float a(int x, int c);` che calcoli il seguente valore

$$\sum_{j=0}^c \frac{X}{j + X}$$

Esercizio 2 (punti 6)

Dato il seguente programma C:

```
#include <stdio.h>
#define N 6

void f(int V[], int p, int *b)
{int i,j;
 p=p-2;
 *b=5;
 for(i=0 ; i<N; i=i+p)
 {j = *b*(i+1);
  V[i]=V[i]+V[j];
 }
 *b = *b + V[++p];
}

void main()
{int V[N]={0,1,2,4,8,16};
 int i, j = -1, b;
 for (i=0; i<N; i=i+j)
  V[i] = V[i] - j++;
 f(V, j, &b);
 for(i=0; i<N; i++)
  printf("%d\t", V[i]);
 printf("\n%d\t%d\n", b, j);
}
```

Cosa viene stampato dal programma? La risposta deve essere opportunamente motivata.
Si spieghi come ci si comporta per l'identificatore `b` definito sia nel main sia nella procedura.
Cosa fa la `#define N 6` all'inizio del programma.

Esercizio 3 (punti 6)

Data la struttura

```
struct bolletta{ char servizio[30];
                int codice;
                int totale;
                int scadenza;    // un numero compreso tra 1 e 365
                }
```

scrivere una procedura che riceve in ingresso un array `B` di strutture `struct bolletta`, la sua dimensione `N` e una data (espressa come giorno dell'anno tra 1 e 365) e scriva su output il codice delle bollette in scadenza prima della data passata come argomento, nonché modifichi l'array ponendo a 0 il campo `totale` di tali strutture. Nel quarto parametro si calcoli il totale da pagare per tutte le bollette in scadenza.

```
void scadenze(struct bolletta B[], int N, int scadenza, int
              *totale);
```

Si scriva poi un possibile main che legga da utente gli elementi del vettore e stampi il vettore modificato dalla procedura, nonché il totale da pagare.

Esercizio 4 (punti 6)

Data la seguente funzione ricorsiva:

```
int ricors(int x, int y)
{if ((x % y == 0))
  return x-y+5;
 else return x+y - ricors(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: `ricors(2,12)`.
Si dica se la funzione è tail ricorsiva e si giustifichi la risposta.

Esercizio 5 (punti 6)

Come gestisce il C le espressioni eterogenee?
Cosa è un vettore e qual è la sua rappresentazione interna?
Regole di visibilità delle variabili.
Cosa è una ALU ?
Cosa sono l'associatività e la priorità tra operatori?

SOLUZIONE

Esercizio 1

```
float a(int x, int c)
{
    if (c == 0) return 1;
    else return x/(c+x*1.0) + a(x,c-1);
}
```

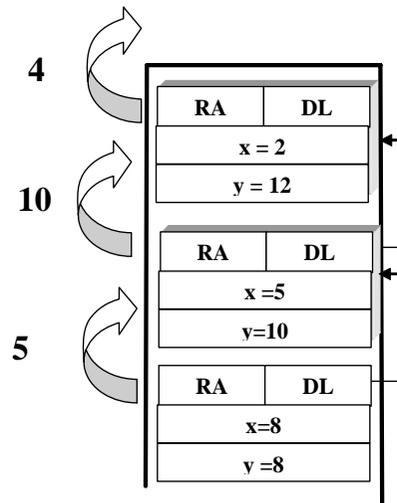
Esercizio 2

Risultato stampato dal programma

```
2  0  4  2  10  32
9  3
```

Esercizio 3

Esercizio 4 Il risultato ottenuto è 4



PROVA SCRITTA DI FONDAMENTI DI INFORMATICA L-A
6 DICEMBRE 2006
COMPITO B

Esercizio 1 (punti 6)

Si scriva una funzione ricorsiva `float b(int x, int c1, int c2);` che calcoli il seguente valore

$$\sum_{j=c1+1}^{c2} \frac{x+c1}{j-c1}$$

Esercizio 2 (punti 6)

Dato il seguente programma C:

```
#include <stdio.h>
#define N 8

void mod(int V[], int p, int *a)
{int i,j;
 *a=N*2;
 for(i=0 ; i<N; i=i+p)
 {j = *a%8;
 V[i]=V[i]+V[j];
 *a= *a-i;
 }
 p = p + *a;
}

void main()
{int S[N]={1,2,3,4,5,6,7,8};
 int i, j = 2, c;
 for (i=0; i<N; i=i+3)
 S[i] = S[i] - i/j;
 mod(S, j, &c);
 for(i=0; i<N; i++)
 printf("%d\t", S[i]);
 printf("\n%d\t%d\n", c, j);
}
```

Cosa viene stampato dal programma? La risposta deve essere opportunamente motivata.
Si spieghi come ci si comporta per l'identificatore `j` definito sia nel main sia nella procedura.
Cosa fa la `#include <stdio.h>` all'inizio del programma.

Esercizio 3 (punti 6)

Data la struttura
`struct studenti{char nome[30];`

```
int mat;
int media;
int esami_mancanti; // un numero compreso tra 1 e 30
}
```

scrivere una procedura che riceve in ingresso un array `S` di strutture `struct studenti`, la sua dimensione `M` una matricola e un voto e modifica l'array calcolando la nuova media dello studente riferito alla matricola passata come argomento (Si supponga che ogni studente abbia 30 esami da sostenere nel curriculum). Il voto da considerare è quello passato come argomento. Si aggiorni anche il campo `esami_mancanti`. Nel quinto parametro si ritorni la variazione della media dello studente.

```
void aggiorna(struct studenti S[], int M, int mat, int voto, int
*Var);
```

Si scriva poi un possibile main che legga da utente gli elementi del vettore e stampi il vettore modificato dalla procedura, nonché il numero di aumenti effettuati.

Esercizio 4 (punti 6)

Data la seguente funzione ricorsiva:

```
int ricors(int x, int y)
{if (!(x % (2*y)))
 return y;
 else return x-y + ricors(x+1,y-1);
}
```

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: `ricors(10,4)`.

Si dica se la funzione è tail ricorsiva e si giustifichi la risposta.

Esercizio 5 (punti 6)

Come gestisce il C gli assegnamenti tra tipi eterogenei?
Cosa è una stringa e qual è la sua rappresentazione interna?
Differenze nei passaggi di parametro per valore e per riferimento.
Cosa sono le memorie di massa e quali tipi abbiamo visto a lezione?
Cosa sono i compilatori e gli interpreti?

SOLUZIONE

Esercizio 1

```
float b(int x, int c1, int c2)
{if (c2 == c1+1) return (x+c1)/(c2*1.0-c1);
 else return (x+c1)/(c2*1.0-c1) + b(x, c1, c2-1);
}
```

Esercizio 2

Risultato stampato dal programma

```
2  2  5  3  9  6  9  8
4  2
```

Esercizio 3

Esercizio 4

Il risultato ottenuto è 16

