

# Esercitazione 9

## Ricorsione

Alberto Marchesi  
Informatica A – Ingegneria Matematica (M—Z)

5 Novembre 2021

Gli esercizi visti a lezione sono segnalati con (\*).

**Esercizio 9.1.** (\*) Scrivere un programma che stampa a video tutti i valori del triangolo di Tartaglia per un certo ordine  $N$ , utilizzando una funzione ricorsiva.

*Esempio:*

1								$N = 0$
1	1							$N = 1$
1	2	1						$N = 2$
1	3	3	1					$N = 3$
1	4	6	4	1				$N = 4$
1	5	10	10	5	1			$N = 5$
1	6	15	20	15	6	1		$N = 6$

**Esercizio 9.2.** (\*) Scrivere una funzione ricorsiva `int convert(int n)` per convertire un intero dato in input nella sua codifica binaria.

**Esercizio 9.3.** (\*) Dato un array di interi, scrivere una funzione ricorsiva che stampi un *triangolo somma* tale per cui il primo livello contiene tutti gli elementi dell'array, e ogni livello successivo contiene come elementi le somme di elementi consecutivi del livello immediatamente precedente.

*Esempio.* Dato  $\{1, 2, 3, 4, 5\}$ , il programma deve stampare (dopo  $\rightarrow$  la spiegazione):

[48]  
[20, 28]  $\rightarrow (20 + 28 = 48)$   
[8, 12, 16]  $\rightarrow (8 + 12 = 20, 12 + 16 = 28)$   
[3, 5, 7, 9]  $\rightarrow (3 + 5 = 8, 5 + 7 = 12, 7 + 9 = 16)$   
[1, 2, 3, 4, 5]  $\rightarrow (1 + 2 = 3, 2 + 3 = 5, 3 + 4 = 7, 4 + 5 = 9)$

**Esercizio 9.4** (TdE 14/11/2008). Si dica cosa stampa il seguente codice, spiegando sinteticamente che funzioni svolgono `f1` ed `f2`. Le risposte che semplicemente spiegano il codice passo per passo non saranno considerate.

```

1  #include <stdio.h>
2
3  int f1(int a,int b){
4  if (b == 0)
5  return 1;
6  return a * f1(a, b-1);
7  }
8
9  int f2(int a,int b){
10 int i;
11 for (i=0; f1(a,i)<=b; i
12 ++);
13 return i-1;
14 }

```

```

1  int main(){
2  int i, a[3] = {2,3,4}, b[3] =
3  {8,30,256};
4  for (i = 0; i<3; i++)
5  printf("%d ", f2(a[i], b[i]));
6  for (i = 0; i < 3; i++)
7  printf("%d ", f1(a[i], f2(a[i], b[i])));
8  for (i = 0; i < 3; i++)
9  printf("%d ", f2(a[i], f1(a[i], b[i])));
10 }

```

**Esercizio 9.5.** Scrivere un programma per calcolare uno zero di una funzione reale in un dato intervallo, implementando il metodo di bisezione in modo ricorsivo. Si assuma che si possa valutare in un punto la funzione reale di cui si vogliono trovare gli zeri, invocando una funzione `double f(double x)`.

**Esercizio 9.6.** (\*) Scrivere una funzione che riceve in input un array di interi e che ordina l'array in modo ricorsivo.

*Suggerimento.* Il metodo più semplice per ordinare un array in modo ricorsivo é noto come **MergeSort**. Dato un array di lunghezza  $N$ , l'idea é di ordinare ricorsivamente le due metà dell'array, utilizzando poi le due metà ordinate per costruire l'array complessivo ordinato, con una semplice operazione detta **Merge**. Il caso base si verifica quando l'array ha lunghezza  $N = 1$ , e quindi, per definizione, é sempre ordinato. Figura 1 riporta un esempio di esecuzione di **MergeSort**.

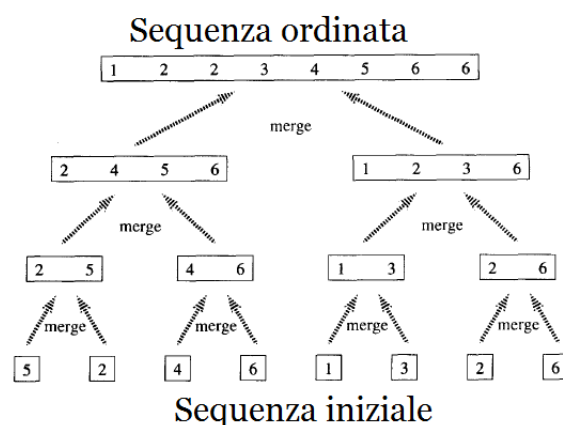


Figura 1: Esempio di MergeSort.

**Esercizio 9.7** (TdE 14/2/2017). Si dica cosa stampa il seguente codice:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <string.h>
4
5 int f(char * x, char * y, int k);
6 int g(int i, char * a, char * b);
7
8 int main() {
9     int i = 3;
10
11     char x[] = "P", y[] = "LO";
12     i++;
13     printf("%c", *x);
14     i = f( x, y, i-1 );
15
16     return 0;
17 }

```

```

1 int f( char * x, char * y, int k ) {
2     if( k ) {
3         k=g(k,x,y);
4     }
5     return k+1;
6 }
7
8 int g(int i, char * a, char * b){
9     int j;
10    if( i%2==0)
11        j=0;
12    else
13        j=1;
14    printf("%c", b[j]);
15    i = f( a, b, i-1 );
16    return i;
17 }

```