

# Corso di Laurea in Matematica

## Insegnamento di Informatica

### Prova in Itinere 3

**a. Dati i seguenti problemi, qualora possibile, per ognuno si realizzi il flow-chart di un possibile algoritmo risolutivo, se ne dia una descrizione in linguaggio lineare e fornirne una programmazione in C. E' richiesto di fare uso di funzioni**

1. Preso in input un valore intero positivo N, produrre in output il triangolo di Floyd con N righe. Si precisa che il triangolo di Floyd è un triangolo rettangolo che contiene numeri naturali, definito riempiendo le righe del triangolo con numeri consecutivi e partendo da 1 nell'angolo in alto a sinistra. Ad es, se  $N = 4$  il corrispondente triangolo di Floyd è:  
1  
2 3  
4 5 6  
7 8 9 10
2. Dato in input un valore intero n, calcolare il relativo numero di Fibonacci, sia in forma iterativa che ricorsiva. Si ricorda che per  $n > 0$ , l'n-esimo numero di Fibonacci (Fib(n)) vale 1 se  $n = 1$  o se  $n = 2$ , altrimenti  $\text{Fib}(n) = \text{Fib}(n-1) + \text{Fib}(n-2)$
3. Dato in input un array di 5 caratteri, produrre in output gli array di tutti i possibili anagrammi.

**b. Specificare la Macchina di Turing che presa in input una sequenza finita di simboli '0' e '1', produca in output il numero di occorrenze di simboli '1', espresso in decimale. Si suggerisce di adottare il simbolo '\$' per indicare l'inizio della sequenza**

**c. Commentare esaustivamente il seguente programma:**

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

void scrivi_lo (void);

int main()
{
    scrivi_lo();

    system("PAUSE");
    return 0;
}

void scrivi_lo (void)
{
    int c;

    if (( c = getchar( ) ) != '\n')
        scrivi_lo ( );
    putchar ( c );
}
```