

# Programmazione

Appello del 10/07/2012

## Esercizio 1 (8 punti)

a. (4 punti) Scrivere l'output del seguente programma Java

```
public class Main {
public static void main(String[] args) {
    System.out.println ( enigma (10,0) );
    System.out.println ( enigma (3,1) );
    System.out.println ( enigma (4,-3) );
    System.out.println ( enigma (-2,10) );
}

public static int enigma (int x, int y){
    if (y< 0) return enigma(-x, -y);
    if (y==0) return 0;
    if (y==1) return x;
    return x + enigma(x, y-1);
}
}
```

b. (4 punti) Riscrivere il metodo enigma in modo equivalente **non** ricorsivo.

## Esercizio 2 (8 punti)

Un array si dice **fortemente crescente** se ogni suo elemento (dal secondo all'ultimo) è almeno pari alla somma di tutti gli elementi che lo precedono.

Scrivere un metodo **public static boolean fc (int [] A)** che preso in input un array A di numeri interi, restituisce **true** se e solo se **A** è "fortemente crescente"

*Ad esempio* se A=[1,1,3,5,15] deve restituire **true**, mentre se A=[1,2,3,5,6] deve restituire **false**.

### Attenzione:

- Per svolgere il compito si hanno a disposizione **90** minuti.
- Scrivere **subito** nome, cognome, matricola e numero del compito su OGNI FOGLIO.
- Durante la prova scritta **non** è possibile abbandonare l'aula.
- Non è ammesso **per nessun motivo** comunicare in qualsiasi modo con altre persone
- **Non** è possibile consultare appunti, libri, dispense o qualsiasi altro materiale.
- Qualsiasi strumento elettronico di calcolo o comunicazione (telefoni cellulari, calcolatrici, palmari, computer, etc...) deve essere **completamente disattivato** e **depositato in vista sulla cattedra**

## SECONDA PARTE

### Esercizio 3 (8 punti)

Si implementino in Java le classi **Prodotto** e **Carrello**.

La classe **Prodotto** ha i seguenti attributi:

- **codice** (una **stringa**)
- **descrizione** (una **stringa**)
- **prezzoUnitario** (un **intero** che esprima il prezzo in centesimi di euro)

ed i seguenti metodi di istanza:

- **costruttore** che crea un oggetto della classe **Prodotto** assegnando descrizione, codice e prezzo unitario (si assuma che il prezzo unitario sia fornito al costruttore in euro, come un parametro di tipo **float**).
- metodi **"get"** per tutti gli attributi, cioè metodi che restituiscono i valori di ciascun attributo (anche in questo caso, il prezzo deve essere restituito in euro, di tipo **float**);
- metodo **public boolean equals (Object o)** che restituisce true se e solo se due prodotti hanno lo stesso codice e lo stesso prezzo unitario (indipendentemente dalla descrizione).

La classe **Carrello** ha i seguente attributi:

- **nome** (una **Stringa**)
- **prodotti** (un **arrayList** di **Prodotto**)
- **quantita** (un **arrayList** di **Integer**)

Gli **arraylist** **prodotti** e **quantita** sono pensati per essere **arrayList** "paralleli", tali che l'**arrayList** **quantita** in posizione **i** contiene il numero di prodotti presenti nel carrello corrispondenti alla posizione **i** dell'**arrayList** **prodotti**.

La classe **Carrello** ha i seguenti metodi di istanza:

- **costruttore** che crea un oggetto della classe **Carrello** prendendo in input il nome del carrello
- metodo **public void addProdotto (Prodotto p, int n)**, che (sfruttando il metodo **equals** della classe **prodotto**)
  - aggiunge il prodotto **p** all'**arrayList** **prodotti** (e corrispondentemente la quantità **n** all'**arrayList** **quantita**), se nel carrello non è già presente il prodotto in questione.
  - altrimenti aggiorna la quantità corrispondente al prodotto **p** incrementandola di **n**.
- metodo **public String getNome ()** che restituisce il nome del carrello.
- metodo **public float getTotal ()** che restituisce il prezzo totale in euro dei prodotti presenti nel carrello.

### Esercizio 4 (8 punti)

- Si scriva una classe astratta **Funzione** per rappresentare funzioni aventi come dominio l'insieme dei numeri interi
- La classe deve avere un metodo astratto **public abstract double calcola (int n)**; che sarà implementato dalle classi che estendono **Funzione** con lo scopo di calcolare il valore della funzione sull'intero **n**.
- La classe deve inoltre implementare un metodo **public int min (int left, int right)** che restituisce uno dei valori del dominio della funzione che realizzano il **minimo** della funzione stessa nell'intervallo **[left, right]**. [si sfrutti il metodo **calcola**]
- Si implementino infine le seguenti classi **concrete** che estendono **Funzione**, ognuna con le opportune variabili di istanza, ed in ognuna delle quali bisogna implementare un **costruttore** ed il metodo **calcola**:
  - La classe **IperboleEquilatera**, il cui costruttore prende quattro **double** (**a,b,c,d**) tali che la funzione rappresentata è  $f(n)=(a \cdot n + b)/(c \cdot n + d)$
  - La classe **FunzionePolinomiale**, il cui costruttore prende un array **A** di **d+1** **double** (dove **d** è il grado del polinomio) tali che la funzione rappresentata è  $f(n)=A[0] + A[1] \cdot n + A[2] \cdot n^2 + \dots + A[i] \cdot n^i + \dots + A[d] \cdot n^d$