

Matricola\_\_\_\_\_ Cognome\_\_\_\_\_ Nome\_\_\_\_\_

## Fondamenti di Informatica

### Prova d'esame del 13/7/2010

#### Regole d'esame:

1. È vietato parlare con altri studenti
2. È vietato consultare appunti, dispense, libri, in qualunque formato
3. È vietato tenere accesi i telefoni cellulari

#### Esercizio 1

Con riferimento al seguente metodo indicare qual è la configurazione dell'input che dà luogo al minor costo di esecuzione (**caso migliore**) e quale quella che dà luogo al maggior costo di esecuzione del metodo (**caso peggiore**). Indicare poi la complessità asintotica di **caso migliore** e di **caso peggiore** in funzione della dimensione **n** dell'array.

```
public static void prova(int[] a){
    for(int i=0;i<a.length;i++)
        if(a[i]%2==0)
            for(int j=i+1;j<a.length;j++)
                a[j]++;
}
```

#### Esercizio 2

Scrivere un metodo di classe ricorsivo che riceve in input un array di int e altri parametri se necessario e restituisce la somma degli elementi dispari dell'array.

#### Note.

1. Una soluzione non ricorsiva non sarà considerata corretta
2. Se ritenuto utile o necessario è possibile scrivere un metodo per l'avvio della ricorsione
3. Il metodo non deve utilizzare variabili che non siano locali

#### Esercizio 3

Siano date le seguenti classi

```
class Rettangolo{

    private double x,y,a,b;

    public Rettangolo(double x,double y,double a,double b){

        this.x=x;
        this.y=y;
        this.a=a;
        this.b=b;
    }

    public double xCentro(){return x;}
```

```

    public double yCentro(){return y;}

    public double altezza(){return a;}

    public double base(){ return b;}

    public String toString(){
        String s="Rettangolo";
        s+="Centro: ("+x+", "+y+")", base: "+b+", altezza: "+a;
        return s;
    }
}

class Quadrato extends Rettangolo {

    public Quadrato(double x,double y,double l){
        super(x,y,l,l);
    }

    public double lato(){return altezza();}

    public String toString(){
        String s="Quadrato";
        s+= "Centro: ("+xCentro()+", "+yCentro()+"), lato: "+altezza();
        return s;
    }
}

```

Si consideri adesso la seguente classe ProvaRettangolo

```

class ProvaRettangolo{

    public static String metodo1(Rettangolo r){
        return r.toString();
    }

    public static String metodo2(Quadrato q){
        return ""+q.lato();
    }

    public static void main(String args[]){

        Rettangolo q=new Quadrato(6,5,2);

        System.out.println("Altezza di q: "+q.altezza());

        System.out.println("Lato di q: "+q.lato());

        System.out.println(metodo1(q));
        System.out.println(metodo2(q));
    }
}

```

Indicare se ci sono errori nel main. Se sì indicare quali sono e spiegare brevemente perché sono errori. Infine, per le istruzioni di stampa che non contengono errori dire che cosa viene stampato.

## Esercizio 4

Con riferimento all'algoritmo di ordinamento **MergeSort** applicato al seguente array, mostrare le diverse attivazioni del metodo **MergeSortRic** e l'effetto di tale attivazione sull'array:

1	2	18	7	19	5	1	6	7	4	5	1
---	---	----	---	----	---	---	---	---	---	---	---

Per rispondere alla domanda si deve indicare, per ogni attivazione, il valore dei parametri **sinistra** e **destra** del metodo **MergeSortRic** e la configurazione dell'array al termine dell'attivazione.

## Esercizio 5

Dire che cosa stampa il seguente codice:

```
class Esercizio{

    public static void main(String args[]){
        int n=7;
        for(int i=0;i<n;i++){
            int a=n/2-i;
            if(a<0){
                a=-a;
            }
            for(int j=0;j<a;j++)
                System.out.print(" ");
            for(int j=a;j<n-a;j++)
                if(i%2==0)
                    System.out.print("+");
                else
                    System.out.print("*");
            System.out.println();
        }
    }
}
```