

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

## Fondamenti di Informatica

### Prova d'esame del 11/6/2013

#### 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

Si consideri il seguente metodo che riceve in input un intero **n**. Indicare, spiegandole brevemente, la complessità asintotica di caso peggiore in funzione di **n**.

```
public static void esercizio(int n){
    int i=0;
    int c=n;
    int a=1;

    while(i<c){
        System.out.println("i: "+i);
        if(i%n==n-1){
            if(a<=n-1){
                a++;
                c=a*n;
            }
        }
        i++;
    }
}
```

#### Esercizio 2

Scrivere un metodo di classe **ricorsivo** che riceve in input un array di caratteri **a** ed altri parametri se necessario, e restituisce una stringa che è la concatenazione dei soli caratteri di **a** in posizione pari. Se, ad esempio, l'array **a** fosse il seguente:

A	B	C	D	E	F	G
---	---	---	---	---	---	---

il metodo dovrebbe restituire la stringa ACEG.

#### 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 C1{
    private int n;
    public C1(int n){
        this.n=n;
    }
    public String metodo1(){
        return ""+n;
    }
    public String metodo2(){
        return ""+(n*2);
    }
}

class C2 extends C1{
    private int b;
    public C2(int n, int b){
        super(n);
        this.b=b;
    }
    public String metodo2(){
        return metodo1()+" "+b;
    }
    public String metodo3(){
        return metodo1()+" "+(b*2);
    }
}
```

Si consideri adesso la seguente classe

```
class C3{
    public static void main(String[] args){
        C1 c1=new C1(10);
        System.out.println(c1.metodo1());
        System.out.println(c1.metodo2());
        System.out.println(c1.metodo3());

        C1 c2=new C2(20,30);
        System.out.println(c2.metodo1());
        System.out.println(c2.metodo2());
        System.out.println(c2.metodo3());

        C2 c3=new C2(40,50);
        System.out.println(c3.metodo1());
    }
}
```

```

        System.out.println(c3.metodo2());
        System.out.println(c3.metodo3());
    }
}

```

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 QuickSort, mostrare l'esecuzione della invocazione **partition(dati,0,11)** essendo **dati** il seguente array di interi:

18	2	17	7	3	20	13	8	4	11	16	19
----	---	----	---	---	----	----	---	---	----	----	----

PER RISPONDERE ALLA DOMANDA SI DEVE MOSTRARE IL VALORE ASSUNTO DI VOLTA IN VOLTA DALLE VARIABILI UTILIZZATE DAL METODO PARTITION E IL CONTENUTO DELL'ARRAY OGNI VOLTA CHE AVVIENE UNO SCAMBIO

#### Esercizio 5

Dire che cosa stampa il seguente codice:

```

class Esercizio{

    public static void main(String args[]){
        int n=6;

        for(int i=0; i<n;i++){
            for(int j=0;j<=i;j++){
                System.out.print("*");
            }
            System.out.println();
        }

        for(int i=0; i<n;i++){
            for(int j=0;j<=i;j++){
                System.out.print(" ");
            }
            for(int j=i+1;j<n;j++){
                System.out.print("*");
            }
            System.out.println();
        }
    }
}

```