

Traccia A

Cognome e Nome:

Quesito 1: Scrivere una classe Java che contenga un metodo main, che richieda all'utente di inserire da tastiera un array di valori interi, e quindi calcoli e stampi a video il minimo e il secondo minimo dei valori pari.

Soluzione

```
public class Quesito1 {
    public static void main(String[] args){
        int min1=0,min2=0;
        Scanner in=new Scanner(System.in);
        System.out.print("Inserire un valore>=2. dimensione=");
        int dim=in.nextInt();
        int[] v=new int[dim];
        int a=0,b=0;
        boolean trovatoPrimoPari=false,trovatoSecondoPari=false;
        \\inserisco l'array e trovo i primi due numeri pari (li chiamo a e b) per inizializzare
        \\minimo e sec. minimo
        for(int i=0;i<dim;i++){
            System.out.println("v["+i+"]=");
            v[i]=in.nextInt();
            if(v[i]%2==0){
                if(trovatoPrimoPari==false){
                    trovatoPrimoPari=true;
                    a=v[i];
                }
                else{
                    trovatoSecondoPari=true;
                    b=v[i];
                }
            }
        }
        if(!trovatoSecondoPari){
            System.out.println("non ci sono due numeri pari nell'array");
            return;
        }
        else{
            if(a<b){
                min1=a;min2=b;
            }
            else{
                min2=a;min1=b;
            }
        }
        for(int i=0;i<dim;i++){
            if(v[i]%2==0){
                if(v[i]<min1){
                    min2=min1;
                    min1=v[i];
                }
            }
        }
    }
}
```

```

        }
        else if((v[i]>min1)&&(v[i]<min2)) min2=v[i];
    }
}
System.out.println("min1="+min1+" min2="+min2);
}
}

```

Quesito 2: Scrivere una classe *Impiegato* che rappresenti *matricola*, *cognome*, *nome* e *retribuzione* di un impiegato, ed una classe *Ufficio* che contenga un array di impiegati. Si dotino entrambe le classi di un costruttore parametrizzato, e si inserisca nella classe *Impiegato* un metodo *media* che accetta in ingresso un array di matricole *M* e calcoli la media delle retribuzioni degli impiegati dell'ufficio che hanno le matricole contenute in *M*.

Soluzione

```

class Impiegato {
    private int matricola;
    private String cognome;
    private String nome;
    private int retribuzione;
    public Impiegato(int m, String c, String n, int r){
        matricola=m;cognome=c;nome=n;retribuzione=r;
    }
    public int getM(){
        return matricola;
    }
    public String getC(){
        return cognome;
    }
    public String getN(){
        return nome;
    }
    public int getR(){
        return retribuzione;
    }
    public Impiegato(Impiegato i){
        matricola=i.getM();
        cognome=i.getC();
        nome=i.getN();
        retribuzione=i.getR();
    }
}

class Ufficio {

```

```

private Impiegato[] imp;
public Ufficio(Impiegato[] im){
    imp=new Impiegato[im.length];
    for(int i=0;i<im.length;i++) imp[i]=new Impiegato(im[i]);
}
public boolean contenuto(int M[],int m){
    for(int i=0;i<M.length;i++) if(M[i]==m) return true;
    return false;
}
public double media(int M[]){
    double media=0;
    int cont=0;
    for(int i=0;i<imp.length;i++){
        if(contenuto(M,imp[i].getM())){
            cont++;media+=imp[i].getR();
        }
    }
    return media/cont;
}
}

```

Quesito 3: Dotare la classe *Ufficio* di un metodo che accetta in ingresso un array di impiegati *A* e restituisce i cognomi di tutti i dipendenti dell'Ufficio che hanno retribuzioni più alte della retribuzione media degli impiegati contenuti in *A*.

\\funzione ausiliaria da inserire nella classe Ufficio

```

public double media(Impiegato A[]){
    double media=0;
    int cont=0;
    for(int i=0;i<A.length;i++){
        media+=A[i].getR();
    }
    return media/A.length;
}

public String[] trova(Impiegato A[]){
    String[] ris;
    int cont=0;
    for(int i=0;i<imp.length;i++){
        if(imp[i].getR(>media(A)) cont++;
    }
    ris=new String[cont];
    cont=0;
    for(int i=0;i<imp.length;i++){
        if(imp[i].getR(>media(A)) {
            ris[cont]=imp[i].getC();
            cont++;
        }
    }
    return ris;
}
}

```

Quesito 4: Qual'è la complessità computazionale dell' algoritmo di ricerca binaria? Darne una spiegazione in breve.

La complessità è $O(\log_2 n)$. L'algoritmo, ad ogni passo della ricorsione, divide per 2 lo spazio di ricerca. Il numero massimo di divisioni possibili è, nel caso peggiore, $\log_2 n$.