

18-03-04-D

```
static boolean verifica(int[] v){
    boolean ok=true;
    int i=1;
    while (ok && i<v.length){
        if (v[i]!=i+v[i-1]) ok=false;
        i++;
    }
    return ok;
}
```

```
public class EsD3_2004_03_18{
```

```
    public static void main(String[] args){
        int dim=Console.readInt("dimensione matrice ");
        int[][] matrice=new int[dim][dim];
        for (int i=0;i<dim;i++){
            System.out.println("elementi riga "+i);
            for (int j=0;j<dim;j++){
                matrice[i][j]=Console.readInt("");
            }
        }
        System.out.println("\nLa matrice e'");
        for (int i=0;i<dim;i++){
            for (int j=0;j<dim;j++){
                System.out.print(matrice[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println("\nRisultato multipli :");
        System.out.println(multipli(matrice));
        System.out.println();
        int h=Console.readInt("r=");
        int k=Console.readInt("c=");
        int[] a=intersezione(matrice,h,k);
        System.out.println("\nRisultato intersezione :");
        for (int i=0;i<a.length;i++){
            System.out.print(a[i]+" ");
        }
        System.out.println();
        int min=Console.readInt("min=");
        int max=Console.readInt("max=");
        int[][] b=filtraColonne(matrice,min,max);
        System.out.println("\nRisultato filtraColonne :");
        for (int i=0;i<b.length;i++){
            for (int j=0;j<b[0].length;j++){
                System.out.print(b[i][j]+" ");
            }
            System.out.println();
        }
        System.out.println();
    }

    static boolean multipli(int[][] m){
        boolean multipli=false;
        for (int i=0;i<m.length;i++){
            for (int j=0;j<m.length;j++){
                if (contieneMultipli(m[i][j],m)) multipli=true;
            }
        }
        return multipli;
    }
}
```

```

}

static boolean contieneMultipli(int x, int[][] m){
    int res=0;
    for (int i=0;i<m.length;i++){
        for (int j=0;j<m.length;j++){
            if (x%m[i][j]==0) res++;
        }
    }
    boolean multipli;
    if (res>1) multipli=true;
    else multipli=false;
    return multipli;
}

static int[] intersezione(int[][] a, int h, int k){
    int[] temp=new int[a.length];
    int pointer=0;
    for (int i=0;i<a.length;i++){
        for (int j=0;j<a.length;j++){
            if (a[h][i]==a[j][k]){
                boolean present=false;
                for (int q=0;q<temp.length;q++){
                    if (temp[q]==a[h][i]) present=true;
                }
                if (!present){
                    temp[pointer]=a[h][i];
                    pointer++;
                }
            }
        }
    }
    int[] res=new int[pointer];
    for (int i=0;i<pointer;i++){
        res[i]=temp[i];
    }
    return res;
}

static int[][] filtraColonne(int[][] m, int min, int max){
    int[][] temp=new int[m.length][m.length];
    int pointer=0;
    for (int j=0;j<m.length;j++){
        boolean elimina=false;
        for (int i=0;i<m.length;i++){
            if (m[i][j]<min || m[i][j]>max) elimina=true;
        }
        if (!elimina) {
            for (int i=0;i<m.length;i++){
                temp[i][pointer]=m[i][j];
            }
            pointer++;
        }
    }
    int[][] res=new int[m.length][pointer];
    for (int i=0;i<m.length;i++){
        for (int j=0;j<pointer;j++){
            res[i][j]=temp[i][j];
        }
    }
    return res;
}
}

```