

18-03-04-B

```
static int[] creaVettore(int[] v){
    int[] s=new int[v.length];
    for (int i=0;i<v.length;i++){
        int somma=0;
        for (int j=i;j<v.length;j++){
            if (v[j]>0) somma+=v[j];
        }
        s[i]=somma;
    }
    return s;
}
```

```
public class EsB3_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 duplicati :");
        System.out.println(duplicati(matrice));
        System.out.println();
        int r=Console.readInt("r=");
        int c=Console.readInt("c=");
        int[][] a=seleziona (matrice,r,c);
        System.out.println("\nRisultato seleziona :");
        for (int i=0;i<a.length;i++){
            for (int j=0;j<a[i].length;j++){
                System.out.print(a[i][j]+" ");
            }
            System.out.println();
        }
        r=Console.readInt("\nr=");
        int[] b=ripetuti(matrice,r);
        for (int i=0;i<b.length;i++){
            System.out.print(b[i]+" ");
        }
    }

    static boolean duplicati(int[][] m){
        boolean duplicati=false;
        for (int i=0;i<m.length;i++){
            for (int j=0;j<m.length;j++){
                if (occurences(m[i][j],m)>1) duplicati=true;
            }
        }
        return duplicati;
    }

    static int occurences(int x, int[][] m){
```

```

    int res=0;
    for (int i=0;i<m.length;i++){
        for (int j=0;j<m.length;j++){
            if (m[i][j]==x) res++;
        }
    }
    return res;
}

static int[][] seleziona(int[][] m, int r, int c){
    int[][] res=new int[2][m.length];
    for (int i=m.length-1;i>=0;i--){
        res[0][m.length-i-1]=m[r][i];
    }
    for (int i=m.length-1;i>=0;i--){
        res[1][m.length-i-1]=m[i][c];
    }
    return res;
}

static int[] ripetuti(int[][] m, int r){
    int[] temp=new int[m.length*m.length];
    int pointer=0;
    for (int i=0;i<m.length;i++){
        for (int j=0;j<m.length;j++){
            if (occurences(m[i][j],m)==r){
                boolean present=false;
                for (int k=0;k<m.length;k++){
                    if (temp[k]==m[i][j]) present=true;
                }
                if (!present){
                    temp[pointer]=m[i][j];
                    pointer++;
                }
            }
        }
    }
    int[] res=new int[pointer];
    for (int i=0;i<pointer;i++){
        res[i]=temp[i];
    }
    return res;
}
}

```