

```

public class e060630 scritto D n1{
    public static int metodo1(int a, int b){
        int s,p=1;
        for (int i = 1; i <= a; i++){
            s=0;
            for (int j = 1; j <= b; j++){
                s=s+i;
            }
            p*=s;
        }
        return p;
    }
    public static void main(String args[]) {
        System.out.println(metodo1(4,2));
        System.out.println(metodo1(3,4));
    }
}

/*
Il metodo1 calcola la seguente espressione (prodotto di somme)
Prodotto(i=1,a) (Sommatoria(j=i,b) (i))
*/

public class e060630 scritto D n2{
    public static int replicati(int[] v){
        int i,j, n=0;
        for (i = 0; i < v.length; i++){
            for (j = 0; j < v.length; j++){
                if (i!=j && v[i]==v[j])
                    break;
            }
            if (j==v.length)
                n++;
        }
        return v.length-n;
    }
    public static void main(String args[]) {
        int[] v={1,2,3,4,5,6,7,8,9};
        System.out.println(replicati(v));
    }
}

public class e060630_scritto_D_n3{

    public static int[][] leggi(){
        int n=Console.readInt("Inserisci la dimensione della
matrice quadrata M > ");
        int[][] M=new int[n][n];
        for (int i=0; i<n; i++)
            for (int j=0; j<n; j++)
                M[i][j]=Console.readInt("Inserisci m["+i+"
["+j+"] > ");
        return M;
    } // fine leggi

    public static void visualizzaMat(int[][] M){
        for (int i=0; i<M.length; i++){
            for (int j=0; j<M[0].length; j++){
                System.out.print("\t"+M[i][j]);
            }
            System.out.println();
        } // fine visualizzaMat

    public static void visualizzaVet(int[] V){
        for (int i=0; i<V.length; i++)
            System.out.print("\t"+V[i]);
        System.out.println();
    } // fine visualizzaVet

    public static boolean colonnaProporzionale(int[][] M, int c1){
        int c2,i;

```

```

        boolean esiste=false;
        for (c2=0; c2<M.length && !esiste; c2++){
            if (c2==c1)
                continue;
            for (i=1; i<M[0].length; i++)
                if ((double)M[0][c2]/M[0][c1]!=(double)M[i][c2]/M[i][c1])
                    break;
            if (i==M.length)
                esiste=true;
        }
        return esiste;
    } // fine colonnaProporzionale

    public static int[] rigaConDistinti(int[][] M){
        for (int i=0; i<M[0].length; i++)
            if (distinti(M[i])==M[0].length)
                return M[i];
        return M[0]; //non sarà mai eseguito esistendo sempre, per
        ipotesi, una rigaConDistinti
    } // fine rigaConDistinti

    public static int distinti(int[] v){
        int i,j, n=0;
        for (i=0; i<v.length; i++){
            for (j=0; j<v.length; j++)
                if (i!=j && v[i]==v[j])
                    break;
            if (j==v.length)
                n++;
        }
        return n;
    }

    public static int[][] sottomatriceC(int[][] M){
        int i, j;
        int[] j_min= new int[M[0].length];
        int min=M[0][0];
        j_min[0]=1;
        for (j=0; j<M[0].length; j++){
            for (i=0; i<M.length; i++)
                if (M[i][j]<min){
                    for (int x=0; x<M[0].length; x++)
                        j_min[x]=0;
                    min=M[i][j];
                    j_min[j]=1;
                }
            else
                if (M[i][j]==min)
                    j_min[j]=1;
        }
        int n=M.length;
        for (int x=0; x<M[0].length; x++)
            n=n-j_min[x];

        int[][] mat=new int[M.length][n];
        int c=0;
        for (j=0; j<M.length; j++)
            if (j_min[j]==0){
                for (i=0; i<M.length; i++)
                    mat[i][c]=M[i][j];
                c++;
            }
        return mat;
    } // fine sottomatriceC

    public static void main(String args[]){
        int[][] M=leggi();
        visualizzaMat(M);

        if (colonnaProporzionale(M,0))
            System.out.println("1- true");
        else

```

```
        System.out.println("1- false");

        int[] V = rigaConDistinti(M);
        System.out.println("2- V=");
        visualizzaVet(V);

        int[][] C = sottomatriceC(M);
        System.out.println("3- C=");
        visualizzaMat(C);
    } // fine main
} // fine class
```