

10-01-05-D

```
public static void stampaVettore(int[] V1, int[] V2)
{
    int minV2 = min(V2);
    for (int i=0; i < V1.length; i++)
        if (V1[i] < minV2)
            System.out.println(V1[i]);
}
```

```
public static int min(int[] V)
{
    int min = V[0];
    for (int i=0; i < V.length; i++)
        if (V[i] < min)
            min = V[i];
    return min;
}
```

```
public class Appello100105_ESE3_D
{
```

```
    public static boolean eSimmetrica(int[][] M)
    {
        if (M.length != M[0].length) return false;
        boolean simmetrica = true;
        for (int i = 0; i < M.length && simmetrica; i++)
            for (int j = i + 1; j < M.length && simmetrica; j++)
                if (M[i][j] != M[j][i])
                    simmetrica = false;
        return simmetrica;
    }
```

```
    public static int[][] moltiplica(int[][] M1, int[][] M2)
    {
        if (M1.length != M2[0].length) return null;
        int[][] P = new int[M1.length][M2[0].length];
        int s = 0;
        for (int i = 0; i < P.length; i++)
            for (int j = 0; j < P[0].length; j++)
            {
                s = 0;
                for (int k = 0; k < M1[0].length; k++)
                    s += M1[i][k]*M2[k][j];
                P[i][j] = s;
            }
        return P;
    }
```

```
    public static int[] trace(int[][] M)
    {
        if (M.length != M[0].length) return null;
        int[] w;
        int[] vett = new int[M.length];
        int count = 0;
        for (int i = 0; i < M.length; i++)
            if (M[i][M.length-1-i] > 0)
            {
                vett[count] = M[i][M.length-1-i];
                count++;
            }
    }
```

```

        if (count == 0)
            return null;
        else
        {
            W = new int[count];
            for (int i = 0; i < count; i++)
                W[i] = vett[i];
        }
        return W;
    }

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

    public static int[][] leggiMatriceQuadrata()
    {
        int n = Console.readInt("Inserisci la dimensione della matrice ");
        int[][] M = new int[n][n];
        for (int i = 0; i < M.length; i++)
            for (int j = 0; j < M[0].length; j++)
                M[i][j] = Console.readInt("Inserisci valore M["+i+""]["+j+"]");
        return M;
    }

    public static void main(String[] args)
    {
        int[][] A = leggiMatriceQuadrata();
        if (eSimmetrica(A))
        {
            System.out.println("La matrice è simmetrica");
            int[][] P = moltiplica(A,A);
            stampaMatrice(P);
            int[] V = trace(P);
            if (V != null)
            {
                System.out.println("Trace");
                for (int i = 0; i < V.length; i++)
                    System.out.println(V[i]+" ");
            }
        }
        else
            System.out.println("La matrice non è simmetrica");
    }

}

```