

10-01-05-A

```
public static boolean metodo2A(int[] V)
{
    boolean alter = true;
    for (int i=1; i < V.length && alter; i++)
        if ((V[i]%2 == 0) && (V[i-1]%2 == 0) || (V[i]%2 == 1) && (V[i-1]%2 == 1))
            alter = false;
    return alter;
}
```

```
public class Appello100105_ESE3_A
{
```

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

```
    public static int[][] costruisciMatrice(int[][] M, int r, int c)
    {
        if (r >= M.length || c >= M[0].length) return null;
        //determiniamo il numero di righe e di colonne della nuova matrice
        int nColonne = M[0].length - c;
        int nRighe = M.length - r;
        int[][] A = new int[nRighe][nColonne];
        for (int i = 0; i < A.length; i++)
            for (int j = 0; j < A[0].length; j++)
                A[i][j] = M[r+i][c+j];
        return A;
    }
```

```
    public static boolean verificaAppartenenza(int[][] M, int[] V)
    {
        boolean appartiene = true;
        boolean presente;
        for (int k = 0; k < V.length && appartiene; k++)
        {
            presente = false;
            for (int i = 0; i < M.length && !presente; i++)
                for (int j = 0; j < M[0].length && !presente; j++)
                    if (M[i][j] == V[k])
                        presente = true;
            if (!presente)
                appartiene = false;
        }
        return appartiene;
    }
```

```
    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 void main(String[] args)
{
    int[][] M = new int[][] {
        {1,2,3,4,1},
        {2,3,4,5,2},
        {3,4,5,6,3},
        {8,9,10,11,8},
        {6,7,8,9,6}
    };
    int[] V = new int[]{1,2,10,11};
    if(eSimmetrica(M))
        System.out.println("La prima colonna è uguale all'ultima");
    else
        System.out.println("La prima colonna non è uguale all'ultima");
    int[][] A = costruisciMatrice(M,3,2);
    System.out.println("Costruisci matrice");
    if (A != null)
        stampaMatrice(A);
    System.out.println("Verifica appartenenza");
    if (verificaAppartenenza(M,V))
        System.out.println("Il vettore appartiene alla matrice");
    else
        System.out.println("Il vettore non appartiene alla matrice");
}

}

```