

Esercizio 1

Il metodo verifica che ogni elemento dell'array y sia divisibile per l'elemento dell'array x che si trova nella posizione opposta. Se i due array hanno dimensioni diverse il metodo restituisce null, altrimenti restituisce un array di booleani di dimensioni pari agli array in input, nel quale l'elemento in posizione i ha valore true se l'elemento dell'array y di indice i è divisibile per l'elemento opposto (ovvero, di indice $x.length-1-i$) di x, e false altrimenti. Nel caso in cui il valore dei parametri sia $x=[3,2,5,4]$ e $y=[8,20,6,9]$ il contenuto dell'array n restituito sarà $[true, true, true, true]$, in quanto $8\%4==0$, $20\%5==0$, $6\%2==0$ e $9\%3==0$.

Esercizio 2

```
public static void testprefisso (String s)
{
    int posizione = s.indexOf('&');
    if (posizione == -1)
        System.out.println ("prefisso non presente");
    else
    {
        String fine = s.substring(posizione+1);
        String inizio = s.substring(0, fine.length());
        if (inizio.equals(fine))
            System.out.println ("prefisso corretto");
        else
            System.out.println ("prefisso scorretto");
    }
}
```

Esercizio 3

```
public class Esercizio3
{
    private static double[][] estrai (double A[][], int i, int j)
    {
        int dim = A.length/2;
        double R[][] = new double[dim][dim];
        for (int k = 0; k < dim; k++)
            for (int h = 0; h < dim; h++)
                R[k][h] = A[k+dim*(i-1)][h+dim*(j-1)];
        return R;
    }

    private static double[][] prodotto (double A[][], double B[][])
    {
        double C[][] = new double[A.length][B[0].length];
        for (int i=0; i<A.length; i++)
            for (int j=0; j<B[0].length; j++)
            {
                C[i][j]=0;
                for (int k=0; k<A[0].length; k++)
                    C[i][j]+=A[i][k]*B[k][j];
            }
        return C;
    }

    private static boolean colonneDispariNulle (double A[][])
    {
        boolean tuttiNulli = true;
        for (int i = 0; i < A.length && tuttiNulli; i++)
            for (int j = 1; j < A[0].length && tuttiNulli; i+=2)
                if (A[i][j] != 0)
                    tuttiNulli = false;
        return tuttiNulli;
    }
}
```

```

private static double[][] differenza (double A[][], double B[][])
{
    double[][] C = new double[A.length][A[0].length];
    for (int i = 0; i < A.length; i++)
        for (int j = 0; j < A[0].length; j++)
            C[i][j] = A[i][j] - B[i][j];
    return C;
}

private static void leggi (double M[][])
{
    for (int i = 0; i < M.length; i++)
        for (int j = 0; j < M[0].length; j++)
            M[i][j] = Console.readDouble ("Elemento in posizione "+i+", "+j+" =");
}

private static void stampa (double 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 dim = 12;
    double A[][] = new double [dim][dim];

    System.out.println ("Lettura della matrice A:");
    leggi(A);

    double A11[][] = estrai (A,1,1);
    double A12[][] = estrai (A,1,2);
    double A21[][] = estrai (A,2,1);
    double A22[][] = estrai (A,2,2);

    double B[][] = differenza (prodotto(A11,A22),prodotto(A12,A21));

    System.out.println ("Stampa della matrice B:");
    stampa (B);

    if (colonneDispariNulle(B))
        System.out.println("Gli elementi delle colonne di indice "+
                           "dispari sono tutti nulli");
    else
        System.out.println("Gli elementi delle colonne di indice "+
                           "dispari NON sono tutti nulli");
}
}

```

Esercizio 4

```

public static boolean verificasomma (int []V1, int []V2)
{
    int sommaPari = 0;
    for (int i = 0; i < V1.length; i+=2)
        sommaPari += V1[i];
    int sommaDispari = 0;
    for (int i = 1; i < V2.length; i+=2)
        sommaDispari += V2[i];
    return (sommaPari == sommaDispari);
}

```