```
public class SoluzioniLab5
{
public static boolean checkColsSumInRange(double m[][],double a,double b)
{
      int i,j;
      double sum;
      boolean ok=true;

      for (i=0; ok && i<m.length; i++) {
            sum=0.0;
            for (j=0; j<m[0].length; j++)
                  sum += m[i][j];
            if ( sum < a || sum > b )
                  ok=false;
            }
      return ok;
      }

public static boolean islowertriangular(double m[][]) {
      int i,j;
      boolean ok=true;

      for (j=0; ok && j<m[0].length; j++)
            for (i=j+1; ok && i<m.length; i++)
                  if ( m[i][j] != 0.0 )
                        ok=false;
      return ok;
      }

public static boolean issimmetric(int mat[][]) {
      int i,j;
      boolean simm=true;

      for (i=0; simm && i<mat.length; i++)
            for (j=i+1; simm && j<mat[0].length; j++)
                  if (mat[i][j] != mat[j][i])
                        simm=false;

      return simm;
      }

public static void transpose(int mat[][]) {
      int i,j;
      int t;

      for (i=0; i<mat.length; i++)
            for (j=i+1; j<mat[0].length; j++) {
                  t = mat[i][j];
                  mat[i][j] = mat[j][i];
                  mat[j][i] = t;
                  }
      }
```

```
public static int intersect(int a[], int b[]) {
      int i=0,j=0,k=0;

      while (i<a.length && j<b.length)
            if ( a[i] < b[j] )
                  i++;
            else if ( a[i] > b[j] )
                  j++;
            else {
                  k++;
                  i++;
                  j++;
            }

      return k;
      }

}
```