

### **Programmazione**

Si scriva un metodo "filtraMatrice" che riceve in input una matrice e restituisce un array contenente tutti gli elementi presenti sulle righe di indice pari che contengono solo elementi dispari.

*Soluzione:*

```
public static int[] filtraMatrice(int[][] M) {
    int numRighe = 0;
    for (int i = 0; i < M.length; i += 2)
        if (rigaTuttiDispari(M, i))
            numRighe++;
    int[] SM = new int[numRighe*M.length];
    int pos = 0;
    for (int i = 0; i < M.length; i += 2)
        if (rigaTuttiDispari(M, i))
            for (int j = 0; j < M.length; j++)
                SM[pos++] = M[i][j];
    return SM;
}

public static boolean rigaTuttiDispari(int[][] M, int riga) {
    for (int j = 0; j < M.length; j++)
        if (M[riga][j] % 2 == 0)
            return false;
    return true;
}
```

## **Programmazione**

Si scriva un metodo "aStrisce" che riceve in input una matrice di interi e restituisce true se e solo se la matrice è *a strisce orizzontali*, cioè se le righe di indice dispari contengono solo zeri, mentre le righe pari non contengono zeri.

*Soluzione con due cicli innestati e variabile booleana di controllo:*

```
public static boolean aStrisce(int M[][]) {
    boolean verificato = true;
    for (int i = 0; i < M.length && verificato; i += 2)
        for (int j = 0; j < M[0].length && verificato; j++)
            if (M[i][j] == 0)
                verificato = false;
    for (int i = 1; i < M.length && verificato; i += 2)
        for (int j = 0; j < M[0].length && verificato; j++)
            if (M[i][j] != 0)
                verificato = false;
    return verificato;
}
```

*Soluzione con due cicli innestati e return anticipato:*

```
public static boolean aStrisce1(int M[][]) {
    for (int i = 0; i < M.length; i += 2)
        for (int j = 0; j < M[0].length; j++)
            if (M[i][j] == 0)
                return false;
    for (int i = 1; i < M.length; i += 2)
        for (int j = 0; j < M[0].length; j++)
            if (M[i][j] != 0)
                return false;
    return true;
}
```

*Soluzione con un solo ciclo innestato e return anticipato:*

```
public static boolean aStrisce2(int M[][]) {
    for (int i = 0; i < M.length; i++)
        for (int j = 0; j < M[0].length; j++)
            if (i % 2 == 0 && M[i][j] == 0 || i % 2 != 0 && M[i][j] != 0)
                return false;
    return true;
}
```