

*Secondo parziale del 21/12/2022 (CdL Informatica + CdL Matematica):
possibili soluzioni di una versione del parziale.*

Esercizio 1. Scrivere un metodo *statico* iterativo che, dati un array bidimensionale di stringhe a ed un array monodimensionale di caratteri c , restituisce un array monodimensionale di booleani b tale che $b[i]$ vale *true* se *tutte* le stringhe nella riga $a[i]$ hanno $c[i]$ come primo carattere, altrimenti $b[i]$ vale *false*.

Esempio: se $a = \{ \{ "adb", "dt", "dir" \}, \{ "jr", "jgt" \}, \{ "fx", "f", "gif", "fy" \} \}$ e $c = \{ 'd', 'j', 'f' \}$, il metodo restituisce l'array $b = \{ false, true, false \}$.

Si assuma che le lunghezze di a e c siano uguali e che ogni stringa in a sia non vuota.

N.B. Occorre passare alla riga successiva *non appena* si trova in $a[i]$ la prima stringa che non soddisfa la condizione data.

```
public static boolean[] m1 (String[][] a, char[] c) {
    boolean[] b = new boolean[a.length];
    for (int i=0; i<a.length; i++) {
        int j=0; boolean ok = true;
        while (j<a[i].length && ok) {
            if (a[i][j].charAt(0) != c[i])
                ok = false;
            j++;
        }
        if (ok) b[i] = true;
    }
    return b;
}
```

Possibile usare $b[i]$ al posto di ok inizializzando $b[i]$ a *true*.

In tal caso il comando condizionale dopo il ciclo interno non è necessario.

Esercizio 2. Scrivere un metodo *statico* iterativo che, dati un array bidimensionale di stringhe a , un array monodimensionale di interi v ed un intero k , restituisce un array monodimensionale di stringhe b tale che $b[i]$ è la stringa ottenuta concatenando le *prime* $v[i]$ stringhe nella riga $a[i]$ (letta da sx a dx) la cui lunghezza è maggiore o uguale a k . Se in $a[i]$ non esistono tali $v[i]$ stringhe, allora $b[i]$ è la stringa ****.

Esempio: se $a = \{ \{ "ops", "t", "cr7", "yr" \}, \{ "qr", "y5", "g" \}, \{ "FM3", "k", "zk" \} \}$, $v = \{ 2, 3, 2 \}$ e $k = 2$, il metodo restituisce l'array $b = \{ "opscr7", "**", "FM3zk" \}$.

Si assuma che le lunghezze di a e v siano uguali e che i valori di v e di k siano strettamente positivi. **N.B.** Occorre passare alla riga successiva *non appena* si trovano in $a[i]$ le $v[i]$ stringhe che soddisfano la condizione richiesta.

```
public static String[] m2 (String[][] a, int[] v, int k) {
    String[] b = new String[a.length];
    for (int i=0; i<a.length; i++) {
        b[i] = "";
        int j=0, cont=0;
        while (j<a[i].length && cont<v[i]) {
            if (a[i][j].length() >= k) {
                b[i] = b[i] + a[i][j];
                cont++;
            }
            j++;
        }
    }
}
```

```

        if (cont < v[i]) b[i] = "**";
    }
    return b;
}

```

Esercizio 3. Scrivere una versione ricorsiva del metodo definito per risolvere l'Esercizio 2 (incluso il requisito del suo **N.B.**).

```

public static String[] m2R (String[] [] a, int[] v, int k) {
    String[] b = new String[a.length];
    return m2R(a,v,k,b,0,0,0);
}

public static String[] m2R (String[] [] a, int[] v, int k, String[] b, int i,
int j, int cont) {
    if (i == a.length) return b;
    if (cont == v[i]) return m2R(a,v,k,b,i+1,0,0);
    if (j == 0) b[i] = "";
    if (j == a[i].length) {
        b[i] = "**";
        return m2R(a,v,k,b,i+1,0,0);
    }
    if (a[i][j].length() >= k) {
        b[i] = b[i] + a[i][j];
        cont++;
    }
    return m2R(a,v,k,b,i,j+1,cont);
}

```

Esercizio 4. Scrivere un metodo *statico* iterativo che, dati un array bidimensionale di interi a ed un array monodimensionale di interi b , restituisce un array bidimensionale di caratteri c tale che l'elemento $c[i][j]$ vale 'y' se $a[i][j]$ è divisibile per $b[i]$, altrimenti $c[i][j]$ vale 'n'. Esempio: se $a = \{\{12,-1\},\{-7,9,3,8\},\{5,10,3\}\}$ e $b = \{4,3,2\}$, il metodo restituisce l'array $c = \{\{'y','n'\},\{'n','y','y','n'\},\{'n','y','n'\}\}$. Si assuma che le lunghezze di a e b siano uguali.

```

public static char[] [] m4 (int[] [] a, int[] b) {
    char[] [] c = new char[a.length] [];
    for (int i=0; i<a.length; i++) {
        c[i] = new char[a[i].length];
        for (int j=0; j<a[i].length; j++) {
            if (a[i][j] % b[i] == 0)
                c[i][j] = 'y';
            else
                c[i][j] = 'n';
        }
    }
    return c;
}

```