Definizione di metodi ricorsivi che restituiscono array

1. (*Prova intermedia del 7/11/2003, Esercizio 3*)
   Scrivere un metodo iterativo che, dati una stringa `s` ed un array `a` di interi, restituisce un array di caratteri `c` tale che l'elemento `c[i]` è il carattere di `s` in posizione `a[i]`. Ad esempio, dati `s = "abcd"` e `a = {2,1,0,1,3}`, il metodo restituisce `c = {'c','b','a','b','d'}`.
   (Nota: si assuma che valga $0 \leq a[i] \leq s.length()-1$ per ogni i.)

```
public static char[] seleziona (String s, int[] a) {
  char[] c = new char[a.length];
  for (int i=0; i<c.length; i++) {
     c[i] = s.charAt(a[i]);
  }
  return c;
}
```

2. (*Prova intermedia del 7/11/2003, Esercizio 4*)
   Scrivere una versione ricorsiva del metodo definito al punto precedente.

```
public static char[] selezionaR (String s, int[] a) {
  char[] c = new char[a.length];
  return selezionaR(s,a,c,0);
}

public static char[] selezionaR (String s, int[] a, char[] c, int i) {
  if (i == c.length) return c;
  c[i] = s.charAt(a[i]);
  return selezionaR(s,a,c,i+1);
}
```

3. (*Prova scritta del 24/03/2004, Esercizio 3 - senza eccezioni*)
   Scrivere un metodo iterativo che, dati un array `a` di stringhe ed un intero `k`, restituisce l'array `a` ruotato di `k` posizioni verso destra.
   Ad esempio, dati `a = {"winnie","pimpi","ro","tigro","kanga"}` e `k = 3`, il metodo restituisce l'array `{"ro","tigro","kanga","winnie","pimpi"}`.

```
public static String[] rotazione (String[] a, int k) {
  int n = a.length;
  String[] b = new String[n];
  for (int i=0; i<n; i++)
    b[(i+k)%n] = a[i];
  return b;
}
```

4. (*Prova scritta del 24/03/2004, Esercizio 4 - senza eccezioni*)
   Scrivere una versione ricorsiva del metodo definito al punto precedente.

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

public static String[] rotazioneR (String[] a, int k, String[] b, int i) {
  if (i == a.length) return b;
  b[(i+k)%a.length] = a[i];
  return rotazioneR(a,k,b,i+1);
}
```

5. (*Prova Intermedia del 28/11/2008, Esercizio 3.*)
   Scrivere un metodo che, dato un array bidimensionale di stringhe `a`, restituisce un array bidimensionale di interi `b` contenente le lunghezze degli elementi di `a` nelle posizioni corrispondenti. Ad esempio, se $a = \{\{$`"abcd","ab","kzz"`$\},\{$`"cde","","hkkhh","a"`$\},\{$`"pprs","lp"`$\}\}$, il metodo restituisce l'array `b` $= \{\{4,2,3\},\{3,0,5,1\},\{4,2\}\}$.

```
public static int[][] lunghezze (String[][] a) {
  int[][] b = new int[a.length][];
  for (int i=0; i<a.length; i++) {
    b[i] = new int[a[i].length];
    for (int j=0; j<a[i].length; j++) {
      b[i][j] = a[i][j].length();
    }
  }
  return b;
}
```

6. Scrivere una versione ricorsiva del metodo definito al punto precedente.

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

public static int[][] lunghezzeR (String[][] a, int[][] b, int i, int j) {
  if (i == a.length) return b;
  if (j == 0) b[i] = new int[a[i].length];
  if (j == a[i].length) return lunghezzeR(a,b,i+1,0);
  b[i][j] = a[i][j].length();
  return lunghezzeR(a,b,i,j+1);
}
```