

Metodi iterativi che restituiscono array

1. (Esame scritto del 4/11/2002, una versione dell'Esercizio 3.)

Scrivere un metodo che, dato un array `a` di interi, restituisce l'array con gli elementi di `a` rovesciati. Ad esempio, dato `a = {5,-7,3,12,1}`, il metodo restituisce l'array `{1,12,3,-7,5}`.

```
public static int[] reverse (int[] a) {
    int[] b = new int[a.length];
    for (int i=0; i<a.length; i++)
        b[i] = a[a.length-1-i];
    return b;
}
```

2. (Esame scritto del 4/11/2002, un'altra versione dell'Esercizio 3.)

Scrivere un metodo che, dati due array `a` e `b` di interi, restituisce l'array in cui `b` è stato attaccato alla fine di `a`. Ad esempio, dati `a = {7,-3,2}` e `b = {-3,4,11,-21}`, il metodo restituisce l'array `{7,-3,2,-3,4,11,-21}`.

```
public static int[] append (int[] a, int[] b) {
    int[] c = new int[a.length + b.length];
    for (int i=0; i<a.length; i++)
        c[i] = a[i];
    for (int i=0; i<b.length; i++)
        c[a.length + i] = b[i];
    return c;
}
```

3. (Esame scritto 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;
}
```

Supponendo che il metodo `seleziona` sia stato definito in una classe `A`, un semplice metodo `main` per provare `seleziona` sull'esempio dato è il seguente:

```

public static void main (String[] args) {
    String s = "abcd";
    int[] a = {2,1,0,1,3};
    char[] c = A.seleziona(s,a);

    for (int i=0; i<c.length; i++)          // stampa dell'array c
        System.out.print(c[i]+" ");
    System.out.println();
}

```

4. (*Esame scritto del 18/9/2002, Esercizio 2.*)

Scrivere un metodo in linguaggio Java che, data una stringa *s* ed un array di caratteri *c*, restituisca un array di interi *conta* tale che *conta[i]* sia il numero di occorrenze del carattere *c[i]* nella stringa *s*. (Esempio: se *s* = "ciao pippo" e *c* = {'a', 'b', 'c', 'p'}, il metodo restituisce *conta* = {1,0,1,3}).

```

public static int[] contaChar (String s, char[] c) {
    int[] conta = new int[c.length];
    for (int i = 0; i < c.length; i++) {
        for (int j = 0; j < s.length(); j++) {
            if (s.charAt[j] == c[i])
                conta[i]++;
        }
    }
    return conta;
}

```

5. (*Esame scritto del 24/3/2004, Esercizio 3.*)

Scrivere un metodo che, dati un array *a* di stringhe ed un intero *k*, restituisce l'array *a* ruotato di *a* 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;
}

```

6. (*Esame scritto del 18/7/2003, Esercizio 3.*)

Scrivere un metodo che, data una stringa *s*, restituisce un array di stringhe contenente tutti i prefissi di *s* in *ordine crescente* (inclusa *s*). Ad esempio, data la stringa "abcdef", il metodo restituisce l'array {"a", "ab", "abc", "abcd", "abcde", "abcdef"}.

```

public static String[] prefissi (String s) {
    int n = s.length();
    String[] a = new String[n];
    for (int i=0; i<n; i++) {
        a[i] = s.substring(0,i+1);
    }
    return a;
}

```

7. (*Esame scritto del 13/12/2002, Esercizio 2.*)

Scrivere un metodo che, dati tre interi i , j ed n , restituisce un array di interi di lunghezza n , i cui primi due elementi sono i e j , ed ogni elemento successivo è uguale al prodotto dei due elementi che lo precedono nell'array. Ad esempio, dati $i=3$, $j=-2$ ed $n=5$, il metodo restituisce l'array $\{3, -2, -6, 12, -72\}$. (Si assuma $n \geq 2$.)

```

public static int[] metodo (int i, int j, int n) {
    int[] a = new int[n];
    a[0] = i;
    a[1] = j;
    for (int k=2; k<n; k++) {
        a[k] = a[k-2] * a[k-1];
    }
    return a;
}

```

Supponendo che `metodo` sia definito nella classe `ArrayProp`, un semplice metodo `main` per provare `metodo` è il seguente:

```

public static void main (String[] args) {
    int i = Integer.parseInt(args[0]);
    int j = Integer.parseInt(args[1]);
    int n = Integer.parseInt(args[2]);
    int[] b = ArrayProp.metodo(i,j,n);
    for (int k=0; k<b.length; k++) {
        System.out.print(b[k] + " ");
    }
    System.out.println();
}

```

8. (*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;
}

```

9. Scrivere un metodo che, dato un array bidimensionale quadrato `a` di caratteri, restituisce l'array bidimensionale di caratteri che rappresenta il triangolo in alto a sinistra inclusa la diagonale. Ad esempio, data la matrice `a`

```

{{'a','b','c'},
 {'d','e','f'},
 {'g','h','i'}}

```

il metodo restituisce l'array `{{'a','b','c'},{'d','e'},{'g'}}`.

```

public static char[][] triangoloA (char[][] a) {
    char[][] b = new char[a.length][];
    for (int i=0; i<a.length; i++) {
        b[i] = new char[a[i].length-i];
        for (int j=0; j<a[i].length-i; j++) {
            b[i][j] = a[i][j];
        }
    }
    return b;
}

```