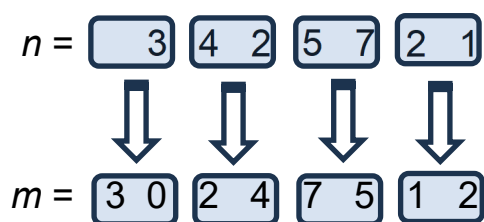


Esercizio n. 1: Inverti sotto-sequenze di un Intero

Sia n un numero intero (di tipo `int`) con $n \geq 0$, e sia k un numero intero (di tipo `byte`) con $k \geq 2$.

La procedura che *inverte* le sotto-sequenze di lunghezza k del numero n produce un nuovo numero intero $m \geq 0$ le cui cifre si ottengono invertendo l'ordine di ogni sottosequenza di k cifre di n partendo da quelle meno significative. Ad esempio, con $n = 3425721$ e con $k=2$ si ottiene il numero $m = 30247512$:



Infatti, si ottiene m partendo dalle cifre meno significative di n , raggruppandole a gruppi di $k=2$ cifre e invertendo l'ordine in cui appaiono le cifre di ciascun gruppo.

Altri esempi partendo dallo stesso $n = 3425721$:

- con $k=2$, $m = 30\ 24\ 75\ 12$
- con $k=3$, $m = 300\ 524\ 127$
- con $k=4$, $m = 2430\ 1275$

Altri esempi:

- se $n = 7777$ e $k=2$, si ottiene $m = 77\ 77$
- se $n = 7$ e $k=3$, si ottiene $m = 700$
- se $n = 20000$ e $k=5$, si ottiene $m = 2$
- se $n = 11000$ e $k=4$, si ottiene $m = 1000\ 0001$
- se $n = 0$ e $k=6$, si ottiene $m = 0$

[CONSEGNA STANDARD]: Scrivere un metodo Java—chiamato `inverti` che, dati in input i due numeri interi n e k come precedentemente definiti, restituisca il numero intero che viene generato dall'applicazione della procedura di inversione delle sotto-sequenze di lunghezza k del numero n come precedentemente descritto.

[CONSEGNA RIDOTTA]: Nella consegna ridotta è possibile svolgere l'esercizio precedente assumendo che le sotto-sequenze possano solo essere di lunghezza 2, ovvero assumendo che $k=2$.

[CONSEGNA EXTRA]: È possibile svolgere la consegna extra solo se si è svolta la consegna standard (non quella ridotta). Due numeri interi n e m sono "collegati" fra loro con passo k se uno dei due numeri corrisponde al risultato prodotto dall'applicazione sull'altro numero della procedura di inversione delle sotto-sequenze di lunghezza k . Ad esempio: i numeri $n=700$ e $m=7$ sono collegati con $k=3$. Scrivere un metodo Java-- chiamato `collegati` che, dati in input i tre numeri interi n , m (di tipo `int`, con $n \geq 0$ e $m \geq 0$) e k (di tipo `byte`, con $k \geq 2$), restituisca `true` se partendo da uno dei due numeri interi (n o m) è possibile produrre l'altro numero attraverso l'applicazione della procedura di inversione delle sotto-sequenze di lunghezza k , e restituisca `false` altrimenti.

NOTA BENE:

- Saranno premiate le soluzioni che occuperanno meno memoria dati.
- È possibile invocare il metodo `Math.pow` per calcolare la potenza di un numero.
- I Junit Test da superare per la consegna ridotta sono quelli della classe **RidottaTest**.
- I Junit Test da superare per la consegna standard sono quelli della classe **StandardTest** (oltre a quelli della classe `RidottaTest`).
- I Junit Test da superare per la consegna extra sono quelli della classe **ExtraTest** (oltre a quelli delle classi `StandardTest` e `RidottaTest` che deve comunque essere superati).
- Nello svolgere l'esercizio NON devono essere utilizzati i metodi `clone`, `arraycopy`, i metodi della classe `Arrays` o `Array` (ad eccezione del metodo `length`, il cui utilizzo è consentito) o i metodi della classe `String` (ad esempio `parseInt()`). L'utilizzo di tali metodi renderà l'esercizio automaticamente insufficiente.

Esercizio n. 2: Diagonali

Sia *arr* un array di interi (di tipo `byte`) a valori positivi (>0) e di dimensione k , con $k>0$. La procedura di “generazione-matrice-con-diagonali” a partire dall’array *arr* consiste nel generare una matrice M di interi (di tipo `byte`) “concatenando” le k matrici quadrate con *diagonale principale e secondaria* costruite a partire dagli elementi dell’array *arr* come descritto di seguito.

In una matrice quadrata, si dice:

- **diagonale principale**, la diagonale che va dall’angolo in alto a sinistra all’angolo in basso a destra;
- **diagonale secondaria**, la diagonale che va dall’angolo in alto a destra all’angolo in basso a sinistra.

La procedura di costruzione delle matrici quadrate con diagonale principale e secondaria può essere descritta come segue:

- Ciascun elemento dell’array *arr* definisce la dimensione della corrispondente matrice quadrata. Se *arr* è composto da k elementi, avrò quindi k matrici quadrate M_0, M_1, \dots, M_{k-1} di dimensione, rispettivamente, $arr[0] \times arr[0], arr[1] \times arr[1], \dots, arr[k-1] \times arr[k-1]$.
- Ciascuna matrice quadrata M_i , con $0 \leq i \leq k-1$, ha tutti valori uguali a zero, ad eccezione degli elementi sulla diagonale principale e su quella secondaria in cui viene copiato il valore di $arr[i]$.

Ad esempio, sia $arr=[2,4,1,3]$, le matrici quadrate con diagonale principale e secondaria M_0 (di dimensione 2×2), M_1 (di dimensione 4×4), M_2 (di dimensione 1×1) e M_3 (di dimensione 3×3) generate a partire dai quattro elementi dell’array sono le seguenti:

Diagonale Principale $M_0 = \begin{pmatrix} 2 & 0 \\ 0 & 2 \end{pmatrix}$ Diagonale Secondaria $M_1 = \begin{pmatrix} 4 & 0 & 0 & 0 \\ 0 & 4 & 0 & 0 \\ 0 & 0 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{pmatrix}$ $M_2 = (1)$ $M_3 = \begin{pmatrix} 3 & 0 & 0 \\ 0 & 3 & 0 \\ 0 & 0 & 3 \end{pmatrix}$

L’applicazione della procedura di “generazione-matrice-con-diagonali” a partire dall’array *arr* genera quindi una matrice M data dalla “concatenazione” delle k matrici quadrate con diagonale principale e secondaria, allineandole alla prima riga e mettendo a zero gli altri elementi.

Nel precedente esempio, la matrice M risultante dalla “concatenazione” delle matrici M_0, M_1, M_2 e M_3 è quindi la seguente:

$$M = \begin{pmatrix} \begin{matrix} M_0 \\ \begin{pmatrix} 2 & 2 \\ 2 & 2 \\ 0 & 0 \\ 0 & 0 \end{pmatrix} \end{matrix} & \begin{matrix} M_1 \\ \begin{pmatrix} 4 & 0 & 0 & 4 \\ 0 & 4 & 4 & 0 \\ 0 & 4 & 4 & 0 \\ 4 & 0 & 0 & 4 \end{pmatrix} \end{matrix} & \begin{matrix} M_2 \\ \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \end{pmatrix} \end{matrix} & \begin{matrix} M_3 \\ \begin{pmatrix} 3 & 0 & 3 \\ 0 & 3 & 0 \\ 3 & 0 & 3 \\ 0 & 0 & 0 \end{pmatrix} \end{matrix} \end{pmatrix}$$

[CONSEGNA STANDARD]:

Scrivere un metodo Java—chiamato `generaMatrice` che, dato un array `arr` di interi (di tipo `byte`) a valori positivi (>0) e di dimensione k , con $k>0$, restituisca una matrice di interi M (di tipo `byte`) generata a partire dall'array `arr` come precedentemente descritto.

[CONSEGNA RIDOTTA]:

Nella consegna ridotta è possibile svolgere l'esercizio precedente (consegna standard) assumendo che `arr` possa contenere solo 1 o 2 elementi, ovvero che $0 < k \leq 2$.

[CONSEGNA EXTRA - facoltativa]:

Scrivere un metodo Java-- chiamato `riduciMatrice` che, data una matrice M generata dalla procedura di "generazione-matrice-con-diagonali" precedentemente descritta, restituisca il corrispondente array `arr` a partire dal quale è stata generata la matrice.

Ad esempio, invocando il metodo `riduciMatrice` sulla matrice M dell'esempio precedente, il metodo dovrà restituire l'array `arr=[2,4,1,3]`.

Nota: La consegna extra può essere svolta solo nel caso in cui sia stata completata la consegna standard, non quella ridotta.

NOTA BENE:

- Saranno premiate le soluzioni che occuperanno meno memoria dati.
- I Junit Test da superare per la consegna ridotta sono quelli della classe **RidottaTest**.
- I Junit Test da superare per la consegna standard sono quelli della classe **StandardTest** (oltre a quelli della classe `RidottaTest`).

- I Junit Test da superare per la consegna extra sono quelli della classe **ExtraTest** (oltre a quelli delle classi StandardTest e RidottaTest che devono comunque essere superati).
- Nello svolgere l'esercizio NON devono essere utilizzati i metodi `clone`, `arraycopy`, i metodi della classe `Arrays` o `Array` (ad eccezione del metodo `length`, il cui utilizzo è consentito) o i metodi della classe `String` (ad esempio `parseInt()`). L'utilizzo di tali metodi renderà l'esercizio automaticamente insufficiente.