

Guía de Ejercicios Semana 9

Introducción

En esta guía trabajaremos con arreglos unidimensionales. Un arreglo es una colección ordenada de datos de largo fijo. Son útiles cuando un problema requeriría utilizar demasiadas variables o cuando no se sabe de antemano cuántas variables se usarán.

En los problemas encontrarás una tabla con ejemplos de entrada y salidas que deben producir tus soluciones. Usa estos ejemplos para verificar que tu solución sea correcta.

Nota: En los ejercicios muchas veces se habla del i -ésimo elemento de una secuencia de números. Al igual que en Java, los índices parten de 0: el 0-ésimo elemento es el primero, el 3-ésimo elemento es el cuarto, etc.

Ejercicios: arreglos

1. Escriba un programa que reciba como input un número N , una secuencia de N números y un número i . El programa debe entregar como output el elemento i -ésimo (partiendo en 0) de la secuencia.

Si el índice i esta fuera de rango (es decir, no existe el elemento i -ésimo) entonces el programa debe imprimir **fuera de rango**.

Entrada	Salida
3 1 2 3 2	3
5 1 2 8 6 5 2	8
5 5 4 3 2 1 6	fuera de rango
1 2 0	2

2. Escriba un programa que sea capaz de determinar si dos secuencias de números son iguales.

El input es un número N_1 , una secuencia de N_1 números, un número N_2 y una secuencia de N_2 números.

El programa debe determinar si ambas secuencias son iguales. Si lo son, debe imprimir **Las secuencias son iguales**. Si no son iguales, debe imprimir **Las secuencias son distintas**.

Entrada	Salida
3 1 2 3 3 1 2 3	Las secuencias son iguales
4 0 3 5 6 4 0 3 5 1	Las secuencias son distintas
3 1 2 3 2 1 2	Las secuencias son distintas
2 1 2 3 1 2 3	Las secuencias son distintas

3. Escriba un programa que busque un número x en una secuencia de números.

El input será un número N , una secuencia de N números, y un número x .

El programa debe entregar como output el primer índice donde se encuentra el elemento x en la secuencia. Si el elemento no se encuentra en la secuencia, el programa debe entregar -1 .

Entrada	Salida
4 1 2 3 4 3	2
5 8 2 5 7 4 8	0
5 4 6 2 3 1 9	-1
3 1 2 3 3	2

4. Escriba un programa que sea capaz de dar vuelta una secuencia de números.

El input será un número N y una secuencia de N números. El programa debe imprimir la secuencia al revés.

Entrada	Salida
4 1 2 3 4	4 3 2 1
5 8 2 5 7 4	4 7 5 2 8
5 4 6 2 3 1	1 3 2 6 4
3 1 2 3	3 2 1

5. Escriba un programa que reciba como input un número N , una secuencia de N números y otra secuencia de N números no repetidos en el rango $[0, N - 1]$.

El programa debe entregar como output los números de la primera secuencia, pero ordenados de la siguiente manera: si el i -ésimo elemento de la segunda secuencia es j , el i -ésimo elemento del output debe ser el j -ésimo elemento de la primera secuencia.

Entrada	Salida
3 1 2 3 0 2 1	1 3 2
4 5 7 2 4 1 2 3 0	7 2 4 5
4 8 6 4 1 0 1 2 3	8 6 4 1
5 1 2 3 4 5 4 3 2 1 0	5 4 3 2 1

6. Escriba un programa que sea capaz de detectar si existen elementos repetidos en una secuencia de números.

El input será un número N y una secuencia de N números.

El programa debe entregar como output si existen números repetidos en la secuencia. Si hay números repetidos, debe imprimir **Hay números repetidos**. En caso contrario, debe imprimir **No hay números repetidos**.

Entrada	Salida
4 1 2 3 4	No hay números repetidos
5 5 7 4 5 1	Hay números repetidos
1 2	No hay números repetidos
6 1 1 2 2 3 3	Hay números repetidos

7. Este ejercicio consta de dos partes:

- a) Escriba un programa que sea capaz de imprimir los 3 elementos menores de una secuencia de números.

El input será un número N y una secuencia de N números, con $N \geq 3$.

El programa debe imprimir el elemento menor, el segundo menor, y el tercer menor, en ese orden.

Entrada	Salida
4 1 2 3 4	1 2 3
5 5 4 4 5 1	1 4 4
3 0 0 0	0 0 0
6 0 2 9 8 7 1	0 1 2

Les puede ser útil recordar que el mayor valor que puede tener un `int` puede ser escrito como `Integer.MAX_VALUE`.

- b) Escriba un programa que sea capaz de imprimir los M menores elementos de un arreglo, con M siendo variable.

El input será un número N , una secuencia de N números y un número M , con $M \leq N$. El programa debe imprimir los M elementos menores en orden, es decir, debe imprimir el elemento menor, el segundo menor, el tercer menor, etc. hasta haber impreso M elementos.

Entrada	Salida
4 1 2 3 4 3	1 2 3
5 5 4 4 5 1 3	1 4 4
4 8 9 6 1 1	1
6 7 9 5 4 7 1 6	1 4 5 7 7 9

Ejercicios Adicionales

1. Escriba un programa que sea capaz de determinar si algún número se repite exactamente M veces en una secuencia de números.

El input será un número N , un número M y una secuencia de N números.

El output del programa deberá ser:

- Si existe algún número x que se repita M veces, entonces el programa debe imprimir **El número $\{x\}$ se repite $\{M\}$ veces**, donde $\{x\}$ y $\{M\}$ deben ser reemplazados por los valores correspondientes.
- Si ningún número se repite M veces, el programa debe imprimir **Ningún número se repite $\{M\}$ veces**, nuevamente reemplazando $\{M\}$ por el valor correspondiente.

Entrada	Salida
5 3 1 2 3 4 5	Ningún número se repite 3 veces
4 2 0 3 7 0	El número 0 se repite 2 veces
7 3 1 4 7 5 1 4 1	El número 1 se repite 3 veces
6 2 1 2 3 4 5 6	Ningún número se repite 2 veces

2. Un conjunto de números se denomina *coprimo* cuando el máximo común divisor entre todos los números es igual a 1. En otras palabras, el número más grande que divide a todos los números del conjunto es 1. Escriba un programa que determine si un conjunto de números es coprimo.

El input será un número N y una secuencia de N números. El output debe ser **Coprimo** si se cumple esta propiedad, o **No es coprimo**. **Máximo común divisor: $\{x\}$** cuando no se cumpla, reemplazando $\{x\}$ por el máximo común divisor del conjunto.

Entrada	Salida
4 2 5 7 11	Coprimo
4 2 6 8 11	Coprimo
5 3 27 12 9 6	No es coprimo. Maximo común divisor: 3
3 51 33 57	No es coprimo. Maximo común divisor: 3

3. Un conjunto de números se denomina *coprimo a pares* cuando el máximo común divisor entre todos par de números es igual a 1. En otras palabras, cada par de números del conjunto es coprimo. Escriba un programa que determine si un conjunto de números es coprimo a pares.

El input será un número N y una secuencia de N números. El output debe ser **Coprimo a pares** si se cumple esta propiedad, o **No es coprimo a pares**. **El par $\{x\}$, $\{y\}$ no es coprimo** cuando no se cumpla, con $\{x\}$ e $\{y\}$ reemplazados por un par de números que no es coprimo.

Entrada	Salida
4 2 5 7 11	Coprimo a pares
4 2 6 8 11	No es coprimo a pares. El par 2, 6 no es coprimo
4 33 10 49 169	Coprimo a pares.
3 44 21 35	No es coprimo a pares. El par 35, 21 no es coprimo