

Práctica 3:

Datos enteros, char, String. Tratamiento de secuencias de datos

Objetivo de la práctica

El objetivo de esta práctica es seguir familiarizándose con los tipos de datos básicos int, char y String y las estructuras de control. Además, empezamos a resolver problemas de tratamiento de secuencia de datos. Una secuencia de datos está formada por cero (secuencia vacía) o un número indeterminado de datos del mismo tipo. Para conocer el final de la secuencia, vamos a usar un valor particular de dicho tipo de dato colocado al final de la secuencia. En este tipo de problemas los esquemas de composición iterativa son fundamentales.

Ejercicios de la sesión práctica obligatorios (sobre 7 puntos)

1. Dos números primos (p , q) son números primos gemelos si están separados por una distancia de 2, es decir, si $q = p + 2$. Escribe un programa Java que muestre las parejas de números primos gemelos menores de 1000.

Los primos Gemelos menores de 1000 son:

(3,5) (5,7) (11,13) (17,19) (29,31) (41,43) (59,61) (71,73)
(101,103) (107,109) (137,139) (149,151) (179,181) (191,193) (197,199) (227,229)
(239,241) (269,271) (281,283) (311,313) (347,349) (419,421) (431,433) (461,463)
(521,523) (569,571) (599,601) (617,619) (641,643) (659,661) (809,811) (821,823)
(827,829) (857,859) (881,883)

2. Una cultura ancestral utilizaba secuencias de números enteros para esconder palabras clave. Si el valor entero de la posición i -ésima de la secuencia es igual a la suma de los valores de las dos posiciones anteriores ($i-1$ e $i-2$), ese valor representa el orden alfabético de una letra. La palabra escondida se forma a partir de la concatenación de todas las letras encontradas en la secuencia. La marca de finalización de la secuencia es el número cero.

A continuación, se muestra un ejemplo de secuencia donde la palabra codificada es "GATO". Los valores de la secuencia que codifican letras son 7 (G), 1 (A), 20 (T), y 15 (O).

6	1	7	3	-2	3	1	5	11	9	20	4	18	-3	15	6	0
		↓				↓				↓				↓		
		G				A				T				O		

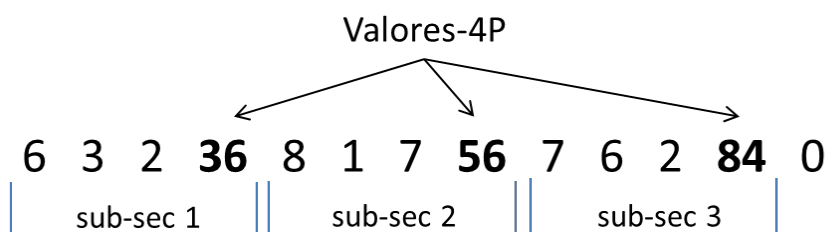
SE PIDE programar en Java un programa que lea de teclado una secuencia de enteros y muestre por pantalla, si la hay, la palabra escondida (en el ejemplo anterior la palabra "GATO").

Ejercicios de la sesión práctica voluntarios (1.5 puntos cada uno)

3. Una *secuencia M4C* satisface las siguientes condiciones:

- Es una secuencia de valores enteros positivos acabada en cero.
- La longitud de la secuencia es múltiplo de 4, sin contar el cero que representa el fin de secuencia.
- Cada sub-secuencia formada por 4 valores cumple que el producto de los tres primeros es igual al cuarto valor. Este cuarto valor recibe el nombre de “valor-4P”.
- Y, los “valores-4P” que aparecen en la secuencia están ordenado de forma creciente.

Por ejemplo, la siguiente secuencia es M4C. Nótese que está formada por 12 enteros (condición 1 y 2), es decir, 3 sub-secuencias de 4 enteros (por ejemplo, la primera sub-secuencia está formada por los valores 6, 3, 2 y 36). Estas sub-secuencias satisfacen la condición 3, y cada una de ellas tiene un valor-4P. En el caso de la primera sub-secuencia, $6 \cdot 3 \cdot 2 = 36$, siendo 36 el “valor-4P”. Además, los 3 valores-4P de la secuencia están ordenados de menor a mayor ($36 < 56 < 84$), satisfaciendo la última de las condiciones.



Se pide programar un programa en Java que lea de teclado una secuencia de enteros acabada en cero y escriba por pantalla si es o no es una secuencia M4C. Por simplicidad, podéis asumir que el usuario siempre introducirá una secuencia de enteros positivos de longitud igual o mayor que 4.

4. Una *Serie Rítmica de Enteros* (SRE) es una secuencia de datos enteros, finalizada con el valor 0, que satisface la siguiente condición:

$$\forall d_i \text{ perteneciente a la serie, } d_i \in [d_{i-1}, d_{i-2}] \text{ o } d_i \in [d_{i-2}, d_{i-1}]$$

Por definición, una secuencia con longitud menor o igual que dos, no es una SRE. Se pide implementar en Java un programa que lea de teclado una secuencia de enteros e imprima por pantalla como resultado si la secuencia leída es o no es una SRE.

Ejemplos:

39 6 21 17 20 18 19 0	ES una serie rítmica de enteros
45 23 34 22 30 0	NO ES una serie rítmica de enteros
34 45 0	NO ES una serie rítmica de enteros
67 0	NO ES una serie rítmica de enteros
0	NO ES una serie rítmica de enteros