

Dept. d'Informàtica i Telecomunicacions	Curs 2016-2017
Grup: DAM1 / DAW1	
M03 Programació I	
Exercicis	
Nom professor/a: Diana Padilla	

07 Exercicis arrays unidimensionals

EXERCICIS BÀSICS ARRAYS

1. Guardar en un array los 20 primeros números pares.
2. Contar el número de elementos positivos, negativos y ceros en un array de 10 enteros.
3. Leer 10 enteros y mostrar la media de los valores negativos y la de los positivos.
4. Leer N alturas y calcular la altura media. Calcular cuántas hay superiores a la media y cuántas inferiores.
5. Leer el nombre y sueldo de 20 empleados y mostrar el nombre y sueldo del empleado que más gana.
6. Leer 10 números enteros por teclado y guardarlos en un array. Calcula y muestra la media de los números que estén en las posiciones pares del array.
7. Leer por teclado la nota de los alumnos de una clase y calcular la nota media del grupo. Mostrar los alumnos con notas superiores a la media.
8. Leer por teclado un número decimal y mostrar su valor en binario.
9. Leer una serie de números por teclado y contar cuántos de ellos acaban en 2.

EXERCICIS ARRAYS

1. Escriure un programa en Java que demani per teclat 10 notes d'alumnes i les mostri per pantalla. Cal utilitzar l'estructura array. **Ex01.java**
2. Escriure un programa en Java que demani 10 notes d'alumnes per teclat, les mostri per pantalla, i també mostri la nota mitja, la nota més alta i la nota més baixa. Cal utilitzar l'estructura array. **Ex02.java**
3. Escriure un programa en Java que demani 15 números per teclat, i mostri per pantalla la suma de tots ells. Cal utilitzar l'estructura array. **Ex03.java**
4. Un palíndrom és una paraula que es llegeix igual d'esquerra a dreta que de dreta a esquerra. Escriure un programa en Java que llegeixi una paraula per teclat, i el programa digui si és un palíndrom o no. Cal utilitzar l'estructura array. **Ex04.java**
5. Escriure un programa en Java que llegeixi de teclat 10 nombres enters, els quals haureu d'emmagatzemar en un array. El programa haurà de comptar el nombre de canvis de signe que hi ha. Un canvi de signe és l'aparició de dos enters de signes diferents, separats com a màxim per zeros. **Ex05.java**
6. Escriure un programa en Java que llegeixi per teclat dos arrays de 10 posicions cadascun. El programa haurà de sumar els dos arrays i mostrar el resultat per pantalla. La manera de sumar dos arrays és sumar cadascuna de les posicions iguals de cada array. Per exemple, la posició 3 de l'array número 1 se sumará amb la posició 3 de l'array número 2, i així amb totes les posicions. **Ex06.java**

7. Escriure un programa en Java que llegeixi per teclat 10 nombres enters, els quals haureu d'emmagatzemar en un array. El programa generarà un segon array que mostrarà per pantalla els 10 nombres ordenats de menor a major valor. **Ex07.java**
8. Escriure un programa en Java que llegeixi per teclat 30 caràcters i els emmagatzemi en un array. Cal que els 30 caràcters es recullin en una variable de tipus String, i convertiu l'String en un array de caràcters. El programa haurà de comptar el nombre de vocals que té l'array. **Ex08.java**
9. Escriure un programa en Java que llegeixi per teclat 30 caràcters seguits que emmagatzemareu en un array i després els posi en un array. El programa haurà de comptar el nombre de vocals que té l'array, indicant el nombre de lletres de cada tipus (nombre de as, nombre de es...). **Ex09.java**
10. Escriure un programa en Java que llegeixi per teclat una frase o conjunt de paraules. Cadascuna de les lletres haurà d'estar emmagatzemada en una posició de l'array. El programa serà un cercador. Haurà de cercar una paraula que introduirà l'usuari per teclat, i indicar per pantalla si s'ha trobat aquella paraula o no. **Ex10.java**
11. Práctica SALTOS DE LONGITUD. Se quiere realizar un programa para gestionar la lista de participaciones en una competición de saltos de longitud. El número de plazas disponibles es ilimitado, es decir, el usuario indicará que no quiere añadir más participantes cuando el número de dorsal sea 0. Sus datos se irán introduciendo en el mismo orden que vayan inscribiéndose los atletas. Diseñar el programa que muestre las siguientes opciones:

1. Inscribir un participante.
2. Mostrar listado de datos.
3. Mostrar listado por marcas.
4. Finalizar el programa.

Si se selecciona 1, se introducirán los datos de uno de los participantes: Número de dorsal (autonumérico según orden de inscripción), nombre, mejor marca del 2013, mejor marca del 2014 y mejor marca del 2015.

Si se elige la opción 2, se debe mostrar un listado por número de dorsal.

La opción 3 mostrará un listado ordenado por la marca del 2015, de mayor a menor.

Tras procesar cada opción, se debe mostrar de nuevo el menú inicial, hasta que se seleccione la opción 4, que terminará el programa.

Se deben utilizar las estructuras que se crean convenientes.

SaltosLongitud.java

12. Práctica SISTEMA SOLAR. Realizar un programa que gestione la información de los planetas del sistema solar. Hay que guardar la siguiente información de un planeta: nombre, masa, diámetro y distancia al Sol. Los datos pueden ser inventados.

Este programa considerará que en el sistema solar hay 9 planetas (Plutón se considera como tal, nosotros no lo vamos a discriminar). Hay que programar el siguiente menú:

1. Consultar planeta.
2. Modificar planeta.
3. Estadísticas.
4. Salir.

En la opción 1 el usuario debe introducir la posición del planeta que quiere consultar, y el programa mostrará todos los datos de éste.

En la opción 2 el usuario indicará el planeta que quiere modificar (introducirá el nombre), y a continuación el programa irá mostrando el nombre anterior y pedirá el nuevo, mostrará la masa anterior y pedirá la nueva, y así.

En la opción 3, el programa mostrará lo siguiente:

- Planeta con más masa.
- Planeta con menos masa.
- Planeta con más diámetro.
- Planeta con menos diámetro.
- Planeta más lejano al Sol.
- Planeta más cercano al Sol.

SistemaSolar.java

13. Práctica CÓDIGO DE BARRAS. En 1952, tres norteamericanos patentaron lo que terminó llamándose código de barras. Consiste en una técnica para representar números (y, en menos ocasiones, letras) mediante una serie de líneas verticales paralelas, con diferentes grosores y separaciones entre ellas. Si bien el primer uso sirvió para identificar de manera automática los vagones de un ferrocarril, hoy los códigos de barras se utilizan en infinidad de lugares, siendo la catalogación de productos la más habitual.

La manera concreta de codificar mediante barras los números y las letras puede ser muy variada, lo que ha llevado a la aparición de diferentes estándares. De todos ellos, el EAN (European Article Number) resulta ser el más extendido. De éste, hay principalmente dos formatos, que se diferencian en el ancho. Existe así el llamado EAN-8, que codifica 8 números, y el EAN-13, que, naturalmente, codifica 13.

6583 9522
EAN-8

8 414533 043847
EAN-13

El último dígito del código se utiliza para detección de errores, y se calcula a partir de los demás. Para eso:

Empezando por la derecha (sin contar el dígito de control que se está calculando), se suman los dígitos individuales, multiplicados por un factor:

- Los dígitos en posiciones impares (empezando a contar por la derecha saltándonos el de control) se multiplican por 3.
- Los dígitos en posiciones pares se multiplican por 1.

Por ejemplo, para el código EAN-8 del ejemplo anterior, la operación a realizar es:

$$2 \cdot 3 + 5 \cdot 1 + 9 \cdot 3 + 3 \cdot 1 + 8 \cdot 3 + 5 \cdot 1 + 6 \cdot 3 = 88$$

El dígito de comprobación es el número que hay que sumar al resultado anterior para llegar a un valor múltiplo de 10. En el ejemplo de EAN-8, para llegar al múltiplo de 10 más cercano por encima del número 88 hay que sumar 2 (y llegar al 90). Ten en cuenta que si la suma resulta ser ya múltiplo de 10, el dígito de control será 0.

Entrada

La entrada estará formada por una serie de casos de prueba. Cada uno contendrá una sucesión de números pertenecientes a un código de barras EAN-8 o EAN-13, incluyendo el dígito de control. Si el número de dígitos es inferior a 8, se asumirá que es un código EAN-8; si es superior a 8 pero inferior a 13, se asumirá EAN-13.

En ambos casos, se completarán el resto de dígitos colocando ceros a la izquierda.

La entrada finalizará con el código especial 0.

Salida

Para cada caso de prueba, el programa indicará si el dígito de control es correcto o no. Si lo es, escribirá SI.

En otro caso, escribirá NO.

También, para cada caso, se deberá indicar, a continuación, si se trata de un código EAN-8 o bien EAN-13.

Entrada de ejemplo

65839522
65839521
8414533043847
5029365779425
5129365779425
0

Salida de ejemplo

SI. EAN-8
NO. EAN-8
SI. EAN-13
SI. EAN-13
NO. EAN-13

Fuente

http://en.wikipedia.org/wiki/European_Article_Number

CodigoBarras.java

14. Práctica SUMA DE NÚMEROS. Todos recordamos cómo aprendimos a sumar, sino más nos vale apagar el ordenador ahora mismo.

Este ejercicio consiste en realizar un programa que emule a un niño de 8 años haciendo una suma. Recordemos el método que seguirá el niño dados dos números de n dígitos:

Empezará cogiendo los dígitos de la derecha de cada número y los sumará entre sí.

En la parte del resultado, apuntará el dígito referente a las unidades del resultado de la suma. En caso de tener dígito de decenas, "se llevará una".

A continuación pasará a los dos dígitos siguientes de cada número, y los sumará. ¡No nos olvidemos que podemos habernos llevado una de la suma de los dígitos anteriores! Si es así, la añadiremos al resultado.

El niño procederá con este proceso hasta que haya sumado todos los dígitos de los números dados.

El programa a realizar debe sumar dos números de n dígitos (no tienen por qué tener el mismo número de dígitos) y mostrarnos el resultado. La suma se tiene que realizar siguiendo el algoritmo del niño de 8 años.

Entrada de ejemplo:

Primer número: 2578921
Segundo número: 78302813

Salida de ejemplo:
El resultado de la suma es: 80881734

SumaNumeros.java

15. Práctica CARRERA DE HORMIGAS. Hemos inventado una nueva prueba de atletismo: la carrera de hormigas.

Consiste en colocar varias hormigas en diferentes posiciones de una fina pista de una determinada longitud. Las hormigas no tienen demasiada libertad de movimientos, por lo que únicamente pueden ir hacia un lado o hacia el otro de la pista. Eso sí, podemos asegurar que cada hormiga correrá por su carril y no chocará con otra.

Cuando se colocan las hormigas en la pista (recordemos que cada una estará en diferentes posiciones), no se sabe en qué sentido comenzará a desplazarse cada una... pero se quiere saber cuáles son el mínimo y el máximo tiempo posible que tardarán todas las hormigas en llegar a uno de los extremos de la pista, sabiendo que todas se mueven a la misma velocidad, de un centímetro por segundo.

Por ejemplo, si la pista tiene una longitud de 10 centímetros, y tenemos hormigas en las posiciones 2, 6 y 7, como mínimo las hormigas tardarán 4 segundos en terminar todas, y como máximo 8.

¿Cuánto tardarán como mínimo y como máximo en completar la carrera 4 hormigas situadas en las posiciones 3, 11, 12 y 13 de una pista de 20 centímetros? ¿Y 8 hormigas en las posiciones 6, 10, 11, 14, 18, 20, 22 y 27 de una de 35?

CarreraHormigas.java