

## Práctica 7: Tipos de datos estructurados. Arrays (2 sesiones)

### Programación 1. Grado en Ingeniería Multimedia

23 de noviembre de 2015

#### Objetivos:

- Conocer la importancia que tienen los arrays como estructuras de datos en el ámbito de la programación.
- Conocer los distintos tipos de arrays.
- Conocer las operaciones posibles a realizar sobre un array.
- Aprender el manejo de arrays bidimensionales.
- Aprender alguno de los métodos de búsqueda y ordenación más comunes en el tratamiento de información mediante el manejo de estructuras de datos de tipo array.

1.

Escribe un programa en C que pida una cadena de caracteres (de máximo 15 caracteres) y devuelva la cadena escrita al revés.

2.

Realiza un programa en C que lea dos palabras, sustituya las letras minúsculas por mayúsculas, nos diga la longitud de cada palabra y las muestre por pantalla en orden alfabético.

Las palabras las deberemos definir de un tamaño fijo, por ejemplo 25.

3.

Realiza un programa que lea desde teclado una frase finalizada en punto. Tamaño máximo 80 caracteres. El programa deberá indicar cuantas mayúsculas y cuántas minúsculas se han introducido.

Para ello, se deberá utilizar un módulo que devuelva el número de mayúsculas a partir del vector dado, y otro módulo que devuelva el número de minúsculas a partir del vector dado.

4.

Escribir un programa que leyendo el día de la semana en que cae el primer día de un mes y el número de días que tiene ese mes, escriba el calendario de ese mes. Por ejemplo, si lee que el día

de la semana es 3 (miércoles) y que el mes tiene 28 días, la respuesta que dé el programa será la siguiente:

Utiliza para ello los módulos que creas adecuados.

Ejemplo de ejecución:

5.

```
¿En qué día de la semana empieza el mes? 3
¿Cuántos días tiene el mes? 28
   1  2  3  4  5
  6  7  8  9 10 11 12
 13 14 15 16 17 18 19
 20 21 22 23 24 25 26
 27 28
```

Escribe en C un programa que calcule la letra del DNI a partir del número de DNI. El programa deberá solicitar el número de DNI y mostrar la letra asociada a ese DNI. Es necesario calcular el resto de dividir el número de DNI entre 23.

La letra del NIF se obtiene a partir de un algoritmo conocido como módulo 23. El algoritmo consiste en aplicar la operación aritmética de módulo 23 al número del DNI. El módulo 23 es el número entero obtenido como resto de la división entera del número del DNI entre 23. El resultado es un número comprendido entre el 0 y el 22. En base a una tabla conocida se asigna una letra. La combinación del DNI con esa letra es el NIF.

```
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22
T R W A G M Y F P D X B N J Z S Q V H L C K E
```

No se utilizan las letras: I, Ñ, O, U

6.

Implementa un programa en C que inicialmente visualice el siguiente menú:

```
MacBook-Pro-de-Rosana:pruebas rosanasatorre$ ./ej
1.- Introduce secuencia de números enteros
2.- Visualiza secuencia en orden descendente
3.- Buscar número
4.- Salir
Opción : █
```

Dependiendo de la opción de menú elegida, se deben realizar las siguientes acciones:

Opción 1: se debe solicitar primero la cantidad de números a introducir y a continuación leer todos los números introducidos (almacenándolos en un array).

Opción 2: se debe ordenar de forma descendente la secuencia de números (ordenar el array) y mostrar por pantalla dicha ordenación.

Opción 3: se debe solicitar inicialmente un número a buscar, mostrándose por pantalla un mensaje que indique si dicho número está o no en la secuencia introducida y otro mensaje indicando el método de búsqueda utilizado.

Opción 4: se debe finalizar la ejecución del programa.

7.

Se define como matriz zig-zag toda matriz en la que se van colocando los números en orden creciente conformando una especie de zigzag,

Realizar un programa que solicite al usuario el número de filas y el número de columnas de la matriz y la rellene en zig-zag. Tanto el número de filas como el de columnas debe estar entre 1 y 15. A continuación debe mostrar la matriz resultante.

Ejemplo de ejecución para una matriz de 3 filas y 4 columnas:

1	6	7	12
2	5	8	11
3	4	9	10

8.

Una sala de cine necesita que le diseñemos un programa en C para poder reservar los asientos de la sala. La sala se compone de 8 filas y cada una de ellas tiene 20 butacas. Del centro de la sala a la derecha encontramos las butacas pares, mientras que del centro a la izquierda encontramos los números impares.

Para realizar una reserva el programa deberá mostrar en primer lugar qué butacas están libres, y preguntar al cliente qué butacas desea. Cuando el cliente solicite una butaca, el programa deberá comprobar si esas butacas están libres, si lo están las reservará sino debe mostrar un mensaje de error.

Ejemplo de ejecución:

```
¿Quiere realizar una reserva?s
PATIO DE BUTACAS
19 17 15 13 11 9 7 5 3 1 2 4 6 8 10 12 14 16 18 20
1
2      * * * * *
3      * * *   * * * *   * * * *
4      * * * * *
5 * * * * *
6      * * *           * * * * *
7      * * * * *   * * * *
8 * * * * *   * * *
¿Qué fila desea?2
¿Qué columna desea?15

Reserva realizada
¿Quiere realizar otra reserva?█
```