Ejercicios recomendados

1. Crear un programa que obtenga del usuario una cadena de caracteres (con un tamaño máximo fijado por el programa) e imprima por pantalla la misma cadena, pero al revés, es decir, empezando desde el último carácter y acabando con el primero.

2. Escribir la sucesión de Fibonacci. El programa debe leer dos números enteros de teclado: el primer dato leído será el número de elementos de la secuencia a mostrar; el segundo, el elemento de la sucesión a partir del que hay que mostrar, es decir, a partir de ese elemento, el programa deberá mostrar tantos elementos de la sucesión como indique el primer número leído. La sucesión de Fibonacci es n0, n1, ..., ni, ..., y se define como sigue:
   - n0 = 0;
   - n1 = 1;
   - ni = ni – 1 + ni – 2.

3. Crear un programa que obtenga del usuario una cadena de caracteres (con un tamaño máximo fijado por el programa) y un carácter, y muestre por pantalla la misma cadena eliminando el carácter indicado.

4. Crear un programa que obtenga del usuario una cadena de caracteres (con un tamaño máximo fijado por el programa) e imprima por pantalla las letras mayúsculas que encuentre en esa cadena.

5. Generar una tabla de equivalencias entre kilómetros y millas náuticas. El programa preguntará el intervalo (por ejemplo, desde 10 a 100 kilómetros) y el paso (por ejemplo, cada 10 kilómetros) y creará una tabla en la que cada línea dará la distancia en millas para cada distancia en kilómetros. Tómese que una milla náutica son 1.852 metros.

6. Calcular el producto escalar de dos vectores. Los vectores tendrán un número de elementos especificado por el usuario (con un máximo fijado por el programa) y obtendrá los elementos de los vectores del usuario.

7. Crear un programa que obtenga del usuario una cadena de caracteres (con un tamaño máximo fijado por el programa) y dos caracteres, y muestre por pantalla la misma cadena pero agregando el segundo carácter solicitado a continuación del primero allí donde este aparezca.

8. Realizar un programa que lea una cadena de caracteres (con un tamaño máximo fijado por el programa) y muestre por pantalla el resultado de cambiar las letras

minúsculas en mayúsculas y viceversa. Si hay otros caracteres que no sean letras, deben salir intactos.

9. Realizar un programa que pida al usuario dos cadenas de caracteres (con tamaños máximos fijados por el programa) y las imprima por pantalla 'mezcladas', es decir, sacando alternativamente un carácter de cada una de ellas. Los caracteres que le sobren a la más larga se imprimirán al final. Por ejemplo, supongamos que las cadenas introducidas por el usuario han sido: tigresa y PATO. El programa debería sacar por pantalla la siguiente cadena: tPiAgTrOesa

10. Crear un programa que obtenga del usuario dos vectores de enteros de igual longitud (pero con una longitud máxima fijada por el programa) y calcule la suma de ambos, rotando el primero un número de posiciones hacia la derecha que indique el usuario. Cada rotación implica que la componente de índice i pasa a ocupar índice i + 1; y que la componente de índice máximo pasa al índice mínimo.

11. El programa obtendrá del usuario una lista de números de longitud arbitraria (pero con una longitud máxima fijada por el programa) y, a continuación, de cada grupo de tres números, intercambiará las posiciones del primero y el tercero, mostrando el resultado por pantalla. Si el último grupo no estuviera completo, lo completará con ceros.

12. Crear un programa que lea de teclado una lista de números. Para ello, primero indicará al usuario el tamaño máximo de la lista y averiguará de él la cantidad de números que el usuario quiere introducir. A continuación, leerá la lista. Una vez leída, el programa mostrará por pantalla otra lista donde el elemento en la posición i es el promedio de los elementos desde la primera posición hasta la i-ésima en la lista original.

13. indique si son primos entre sí. Nota: dos números son primos entre sí si no comparten ningún divisor, es decir, si su máximo común divisor es 1. Por ejemplo, 12 y 5 lo son, mientras que 10 y 6 no lo son.

14. Realizar el sumatorio de $1/n^2$ desde n = 1 hasta el máximo que el usuario especifique por teclado. El programa pedirá al usuario el valor máximo de n y dará como salida la suma total de los valores $1/1^2 + 1/2^2 + 1/3^2 + ... + 1/n^2$. Nota: Las matemáticas nos dicen que, si n es suficientemente alto, ese valor se aproxima a $\pi^2/6$ (≈ 1,64493).

15. Crear un programa que sume los elementos de la sucesión de Padovan entre dos posiciones indicadas por el usuario (ambas inclusive). La sucesión de Padovan p0, p1, p2,... se define como:
    - p0 = p1 = p2 = 1;
    - pn = pn – 2 + pn – 3.

16. 1. Evaluar un polinomio de grado arbitrario en un punto x0 especificado por el usuario. Los coeficientes y el grado del polinomio también serán especificados por el usuario.

17. Calcular la suma de dos matrices. Las matrices tendrán unas dimensiones máximas fijadas por el programa. El programa solicitará al usuario las dimensiones de las matrices, comprobará que no superan los límites fijados y mostrará la matriz resultado fila por fila.

18. Extraer la diagonal principal de una matriz. Se mostrará la matriz original, fila por fila y, después, la diagonal, en una única lista. Las dimensiones máximas de la matriz se fijarán por el programa. El programa debe asegurarse de que el número de elementos no supera el espacio disponible.

19. Tomando como entrada una cadena de texto (de longitud máxima fijada), eliminar las consonantes y mostrar ambas cadenas: la original y la cadena con las consonantes eliminadas.

20. Crear un programa que obtenga del usuario una cadena de caracteres (de longitud máxima fijada) y sustituya en ella la vocal más frecuente en ella por la menos frecuente en ella (si hay varias que son la menos frecuente, se elegirá una de ellas de forma arbitraria).

21. Dados una matriz y un vector, calcular el vector que se obtiene al multiplicar la matriz por el vector. Mostrar únicamente el vector resultado. Las dimensiones máximas de la matriz y del vector se fijarán por programa. El programa debe asegurarse de que el número de elementos no supera el espacio disponible.

22. Mostrar una lista con los números primos que se encuentran entre dos números dados.

23. Hacer un histograma de las vocales de un texto, es decir, indicar cuántas veces aparece cada vocal en ese texto. Se leerá una cadena de caracteres (de tamaño máximo fijado) y se mostrará el número de veces que aparece cada vocal en él. El programa deberá asegurarse de que el número de caracteres leídos no excede el espacio disponible.

24. Contar el número de palabras de un texto. Se leerá una cadena de caracteres (de tamaño máximo fijado) y se mostrará cuántas palabras tiene. El programa ha de comprobar, en primer lugar, que el texto contiene únicamente letras, espacios en blanco y signos de puntuación (se consideran signos de puntuación válidos la coma, el punto y coma, el punto y los dos puntos) y, si no es así, debe terminar de inmediato dando un error. Si la cadena es correcta, debe devolver el número de palabras de que consta, teniendo en cuenta que cualquier espacio en blanco

o signo de puntuación separa palabras y que estos separadores podrían estar repetidos.

25. Llevar un histograma de resultados del evento “tirar un dado”. Un histograma es una lista que cuenta, para cada elemento, el número de veces que ha salido ese resultado. Por ejemplo, la lista (el histograma): 5,3,7,2,6,1 indicaría que el 1 ha salido cinco veces, el 2 tres veces, el 3 siete veces, y así sucesivamente. Para ello, utilizar la función random y asociadas, presente en la biblioteca stdlib.h (Ver nota (*)).

26. Repetir el ejercicio anterior, pero simulando la tirada de dos dados (tener en cuenta que ahora los posibles resultados son los números comprendidos entre 2 y 12).

27. Dibujar en pantalla una pirámide construida con el carácter ‘*’. El programa leerá, en primer lugar, los pisos que tiene la pirámide, y la mostrará centrada sobre su base. Ejemplo: 4 pisos:

```
   *
  ***
 *****
*******
```

28. Visualizar un entero en binario codificado en ASCII (Ver nota (**)). Se visualizarán tanto el entero leído como el resultado de su conversión. Nota: la función printf no dispone de esa conversión.

29. 15. Extraer la diagonal secundaria de una matriz. Se mostrará la matriz original, fila por fila y, después, la diagonal secundaria, en una única lista. Las dimensiones máximas de la matriz se fijarán por el programa. El programa debe asegurarse de que el número de elementos no supera el espacio disponible.

30. Imprimir el primer carácter de cada palabra que exista en una cadena de caracteres de tamaño máximo fijado. El programa ha de comprobar, en primer lugar, que el texto contiene únicamente letras, espacios en blanco y signos de puntuación (se consideran signos de puntuación válidos la coma, el punto y coma, el punto y los dos puntos) y, si no es así, debe terminar de inmediato dando un error. Si la cadena es correcta, el programa imprimirá en la línea siguiente el primer carácter de cada palabra, separándolos entre sí por un espacio.

NOTAS:

(*) Las funciones random y srandom, de la librería stdlib.h, permiten generar números pseudo-aleatorios. Cada vez que se invoca, random devuelve un entero pseudo-aleatorio comprendido entre los valores 0 y (231 − 1). Los números los obtiene de una secuencia generada por medio de un algoritmo que hace que parezcan aleatorios. Sin embargo, esto no es realmente así. Cada vez que se ejecute un programa, la secuencia de números que devuelve la función random es la misma. Para evitar esto, se debe usar la función srandom al inicio del programa. Esta función permite establecer la "semilla", es decir, la posición de la secuencia por la que se empiezan a generar los números. Recibe un parámetro, que es un número a partir del cual obtiene la posición inicial de la secuencia, de modo que, si cada vez que se ejecuta el programa, la función srandom utiliza un número diferente como "semilla", los números que irá generando la función random serán diferentes. Una forma de que el número que se utiliza como semilla sea diferente en cada invocación del programa es utilizar la función time, de la librería time.h, que devuelve un entero con el tiempo en segundos desde las 0 horas, 0 minutos y 0 segundos desde el 1 de enero de 1970. Dado que esta cuenta es diferente cada vez que se ejecute el programa, se puede utilizar este valor como semilla para la función srandom. Evidentemente, para la depuración del programa lo interesante es que la secuencia sea precisamente siempre la misma, por lo que, en este caso, el valor que se usa como semilla deberá ser siempre el mismo.

(**) El binario codificado en ASCII es un código que representa cada bit del dato por un carácter ASCII, bien el '0', bien el '1', dependiendo del valor del bit. De esta manera, un número entero se representa por una cadena de caracteres ASCII. Por ejemplo, el número decimal 5, en binario puro de 8 bits, se representaría así: 00000101. Por lo tanto, en binario codificado en ASCII, se utilizaría un código ASCII para cada bit del número:

'0' '0' '0' '0' '0' '1' '0' '1' '0',

donde '0' representa el código ASCII del 0 y '1' el del 1.