Relación de ejercicios tema 1: Arrays, cadenas estilo C y matrices

Metodología de la Programación, 2015-2016

1 Problemas básicos

1. ¿Es correcto el siguiente programa para calcular la media de una serie de 10 elementos? En caso de no serlo, indica cuál es el problema.

```
#include <iostream>
using namespace std;

int main() {
  int datos[10];

  for (int i=0; i<=10; i++)
    cin>> datos[i];

  int suma=0;
  for (int i=0; i<=10; i++)
    suma += datos[i];

  cout<< "La suma es: " << suma << endl;
}
```

2. ¿Es correcto el siguiente programa para calcular la varianza? En caso de no serlo, indica cuál es el problema.

```
#include <iostream>
#include <cmath>
using namespace std;

int main() {
  int datos[10];

  for (int i=0; i<10; i++)
    cin >> datos[i];

  int suma=0;
  for (int i=0; i<10; i++)
    suma+= datos[i];

  double media = suma/10;

  double suma2=0;
  for (int i=0; i<100; i++)
    suma2 += pow(datos[i]-media,2);

  double varianza=suma2/10;
  cout<< "Varianza: " << varianza<< endl;
}
```

3. Haz un programa que lea un conjunto de números enteros (como máximo 100), los guarde en un array e indique cuál es el mayor de todos. Tanto la lectura de los elementos del array como el cálculo del máximo se harán mediante funciones. Debe incluirse un programa principal (**main**) para probar las funciones implementadas.

4. Haz un programa que lea una serie de números enteros (como máximo 100), los guarde en un array y que lo invierta, mostrando el resultado. Si la secuencia introducida es **1, 100, 34, 48,**

53, entonces el resultado final mostrado debe ser **53, 48, 34, 100, 1**. Deberá contener funciones para: lectura de los elementos del array, mostrar los elementos del array en la salida estándar e invertir los elementos de un array. Se implementará la función **main** para probar que todo funciona de forma correcta.

5. Haz un programa que lea una serie de números enteros (como máximo 100) y que elimine los elementos repetidos guardándolos en un nuevo array. Debe contener funciones para leer los elementos de un array, imprimirlo y eliminar los elementos repetidos. Haz una segunda versión de la función para eliminar los elementos repetidos de forma que la eliminación de repetidos se haga sobre el mismo array de entrada, sin usar un segundo array. Las funciones implementadas se probarán en la función **main**.

6. Haz un programa que obtenga la mayor secuencia monótona creciente de un array de enteros, guardándola en otro array (mediante una función). El array se inicializará de forma explícita en la función `main`. El programa contendrá además una función para imprimir un array y así poder mostrar el resultado.

7. Haz un programa que lea un texto desde `cin` y que, al finalizar la entrada de datos, muestre cuántas veces aparece cada letra (a..z), teniendo en cuenta lo siguiente:

   - la entrada de datos finaliza con el carácter `#`.
   - no se diferenciarán mayúsculas y minúsculas.
   - el texto se lee carácter a carácter en una función que guarda los caracteres en un array de `char` recibido como parámetro.
   - usaremos una función que reciba como parámetro un array de caracteres y construya un array de enteros con la frecuencia para cada carácter.

8. Haz un programa que contenga una función que reciba un primer parámetro de tipo cadena de caracteres (estilo C) y que devuelva una copia de la misma en un segundo parámetro.

9. Haz un programa que contenga una función que compruebe si una cadena (estilo C) es o no un palíndromo, es decir, que se lee igual de izquierda a derecha que de derecha a izquierda sin tener en cuenta los espacios. Por ejemplo, `anilina` sería un palíndromo. La función para ver si la cadena es un palíndromo, se ayudará de otra función auxiliar que, dada una cadena, obtiene otra donde se han eliminado los espacios en blanco.

10. Haz un programa que contenga una función que reciba como entrada dos cadenas de caracteres estilo C (cad y subcad). Debe comprobar si la segunda (subcad) está presente o no dentro de la primera (cad). Si se encuentra la subcadena, devolverá la posición donde se encuentra y, en caso contrario, devolverá -1.

11. Haz un programa que calcule, mediante una función, la traza (suma de los elementos de la diagonal principal) de una matriz 2D.

12. Haz un programa que calcule la traspuesta de una matriz 2D. La trasposición se hace por medio de una función.

13. Haz un programa que sume dos matrices de enteros de tres dimensiones, usando una función auxiliar para realizar la suma.

14. Supongamos que se eligen dos puntos de forma aleatoria (con distribución uniforme) en el cuadrado unitario $[0, 1]^2$. Escribid un programa que estime la longitud media esperada del

segmento que une dichos puntos. Hacedlo de forma que los resultados de cada prueba (cada selección de par de números) se almacenen en un array. La implementación debe ser lo más modular posible, incluyendo funciones auxiliares necesarias para todas las tareas que se consideren oportunas.

15. Escribid un programa que rellene dos arrays a y b con enteros, obtenidos de acuerdo a la siguiente recurrencia:

$$\begin{aligned} a_0 = b_0 &= 1 \\ a_n &= b_{n-1} \\ b_n &= a_{n-1} + 2b_{n-1} \end{aligned}$$

El programa debe generar una tabla en la que cada fila tendrá la siguiente información:

$$k \;\; a_k \;\; b_k \;\; a_k/b_k$$

Haz una conjetura sobre el valor de $\lim_{n->\infty} \frac{a_n}{b_n}$. EL código resultante debe estar modularizado, incluyendo funciones auxiliares para todas las tareas que se consideren oportunas

2 Problemas complementarios

En los siguientes enunciados se debe interpretar que las palabras **vector** y **array** se usan de forma intercambiable.

1. Escribir una definición apropiada de vectores para cada uno de los siguientes problemas:
   (a) Definir un vector de 12 componentes enteros llamado `vector`. Asignar los valores $1, 4, 7, 10, \ldots, 34$ a los elementos del vector.
   (b) Definir un vector de cuatro caracteres llamado `letras`. Asignar los valores 'H', 'o', 'l', 'a' a las componentes del vector.
   (c) Definir un vector de 6 elementos llamado `valor`. Asignar los valores 1, $\frac{1}{2}$, ..., $\frac{1}{6}$ a los elementos del vector.
2. Dado un vector de números reales, realizar una función que determine el primer y segundo elemento más grandes del vector.
3. Realizar una función que compruebe si dos vectores de caracteres son iguales.
4. Realizar una función que elimine los elementos repetidos de un vector, guardando el resultado en el mismo vector.
5. Realizar una función que mezcle de forma ordenada dos vectores (que ya están ordenados) y que devuelva un tercer vector como resultado.
6. Construir una función que permita ordenar de forma descendente los elementos de un vector.
7. Se pide construir una función que tenga como entrada un vector de caracteres (vector con tipo de dato base `char`), y suprima todas las secuencias de espacios en blanco de longitud mayor de uno. Por ejemplo, si el vector original es `(' ','a','h',' ',' ',' ','c')`, el vector resultante debe ser `(' ','a','h',' ','c')` Las modificaciones se harán sobre el mismo vector de entrada y no se podrán usar vectores auxiliares. Debe hacerse lo más eficiente posible.

8. Construir una función que dada una cadena de caracteres que contiene un número entero, devuelva su correspondiente valor numérico. Razonar sobre qué hacer si la cadena no contiene un número entero.

9. Construir una función que dada una cadena de caracteres `cad` y dos valores enteros `pos` y `tam` modifique `cad` eliminando los `tam` caracteres empezando en la posición `pos` de `cad`. Razonar sobre todos los casos posibles que pueden suceder.

10. Construir una función que inserte una cadena de caracteres dentro de otra cadena en una determinada posición.

11. Construir una función que ordene alfabéticamente un vector de cadenas de caracteres.

12. Realizar una función que lea un número natural y lo traduzca a morse. Se usa raya (**-**) para codificar la señal larga y punto (**.**) para la señal corta. Los números se codifican de la forma siguiente:

    `0 ----- 1 .---- 2 ..--- 3 ...-- 4 ....- 5 ..... 6 -.... 7 --... 8 ---.. 9 ----.`

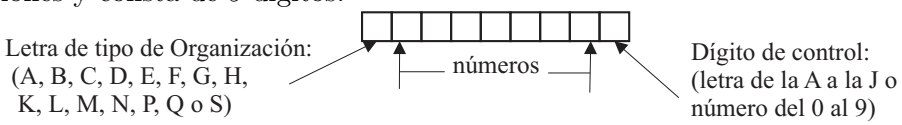
    De esta forma, cada número en morse se representará en C++ por una cadena de 5 caracteres compuesta de puntos y rayas, según la tabla anterior. Un ejemplo de lo que debe hacer el programa es:

    ```
    Introduzca un número natural: 2005
     ..--- ----- ----- .....
    Introduzca un número natural: 1992
     .---- ----. ----. ..---
    ```

13. Construir una función que reciba una cadena de caracteres y compruebe si contiene un número de identificación fiscal (NIF) o número de identidad de extranjero (NIE) válido. La función debe devolver el tipo de documento y si es o no válido. Decidir cómo se devuelve esta información.

    El NIF se forma con 8 dígitos seguidos de una letra (*dígito de control*). El procedimiento empleado para el cálculo del NIF consiste en hallar el resto de la división por 23 del número formado por los 8 dígitos. El resto resultante (comprendido entre 0 y 22) se corresponde con la letra en dicha posición (comenzando por 0) de la secuencia `TRWAGMYFPDXBNJZSQVHLCKE`. El NIE sigue el mismo procedimiento pero su sustituyendo el primer dígito del NIF por la letra X.

14. El CIF (Código de Identificación Fiscal) es un elemento de identificación administrativa para organizaciones y consta de 9 dígitos:

    El primer dígito es una letra que indica el tipo de la organización y puede ser una de los siguientes:

    A - Sociedades Anónimas. B - Sociedades de Responsabilidad Limitada. C - Sociedades Colectivas. D - Sociedades Comanditarias. E - Comunidades de Bienes. F - Sociedades Cooperativas. G - Asociaciones y otros tipos no definidos. H - Comunidades de propietarios en régimen de propiedad horizontal. N - Entidades no residentes. P - Corporaciones Locales. Q - Organismos Autónomos, estatales o no, y asimilados, y Congregaciones e Instituciones religiosas. S - Órganos de la Administración del Estado y Comunidades Autónomas.

    Los siete dígitos siguientes son números y el último es el *dígito de control* que puede ser una letra (en caso del CIF de Entidades no residentes (N), Corporaciones Locales (P), Organismos

Autónomos (Q) ó Organismos de la administración (S)) o un número (en el caso de CIF de las restantes Sociedades).

Las operaciones para calcular el dígito de control se realizan sobre los siete dígitos centrales y son las siguientes:

(a) Sumar los dígitos de la posiciones pares. Suma = A

(b) Para cada uno de los dígitos de la posiciones impares, multiplicarlo por 2 y sumar los dígitos del resultado.
Ejemplo: (8 * 2 = 16 $\longrightarrow$ 1 + 6 = 7)
Acumular el resultado. Suma = B

(c) Sumar A + B = C

(d) Tomar sólo el dígito de las unidades de C y restárselo a 10. Esta resta nos da D.

(e) A partir de D ya se obtiene el dígito de control. Si ha de ser numérico es directamente D y si se trata de una letra se corresponde con la relación: A = 1, B = 2, C= 3, D = 4, E = 5, F = 6, G = 7, H = 8, I = 9, J = 10

Ejemplo para el CIF: Q1818009

(a) Utilizamos los siete dígitos centrales = 1818009

(b) Sumamos los dígitos pares: A = 8 + 8 + 0 = 16

(c) Posiciones impares:
1 * 2 = 2 $\rightarrow$ 2
1 * 2 = 2 $\rightarrow$ 2
0 * 2 = 0 $\rightarrow$ 0
9 * 2 = 18 $\rightarrow$1 + 8 = 9
Sumamos los resultados: B = 2 + 2 + 0 + 9 = 13

(d) Suma parcial: C = A + B = 16 + 13 = 29

(e) El dígito de las unidades de C es 9. Se lo restamos a 10 y nos da: D = 10 - 9 = 1

(f) Como el dígito de control ha de ser una letra es la "A".

Escribir una función que reciba una cadena de caracteres y devuelva si se trata de un CIF válido.

15. Como parte de un programa para imprimir el índice de un libro, se necesita una función que imprima cada línea. La función toma dos palabras almacenadas en cadenas de caracteres y la longitud de una línea `n` como un entero y escribe ambas palabras en una sola línea. Ambas palabras estarán separadas por puntos de tal forma que la longitud total de la línea sea `n` caracteres.

    Ejemplo:

    Primera palabra: `tortuga`
    Segunda palabra: `153`
    Longitud de la línea: 30
    `tortuga.....153`

16. Dadas dos cadenas de caracteres, construir una función en C++ que devuelva las veces que aparece la segunda cadena dentro de la primera. Reflexionar sobre qué debería devolver la función anterior, si las dos cadenas de entrada son `"aaaaa"` y `"aaa"`.

17. Responder apropiadamente a las siguientes cuestiones:

(a) Definir una matriz bidimensional 3×4 de enteros llamada `mat`. Asignar los siguientes valores a los elementos de la matriz.

$$\begin{array}{cccc} 10 & 12 & 14 & 16 \\ 20 & 22 & 24 & 26 \\ 30 & 32 & 34 & 36 \end{array}$$

(b) Definir una matriz bidimensional 3×4 de enteros llamada `mat`. Asignar los siguientes valores a los elementos de la matriz.

$$\begin{array}{cccc} 10 & 12 & 14 & 16 \\ 20 & 22 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{array}$$

(c) Definir una matriz tridimensional $10 \times 10 \times 10$ de enteros. Asignar en cada posición (i, j, k) la suma de los tres índices.

18. Realizar un programa modular que rellene una matriz bidimensional 7×5 de enteros con datos aleatorios e imprima por pantalla el número mayor y menor de la matriz y sus posiciones (fila y columna).

19. Construir una función para comprobar si dos matrices son iguales.

20. Construir una función para comprobar si una matriz es simétrica.

21. En un congreso cuya duración es de 6 días, tienen lugar conferencias en 5 salas. Se desea saber:

    (a) El total de congresistas que asisten a cada una de las salas.
    (b) El total de congresistas asistentes cada día del congreso.
    (c) La media de asistencia a cada sala.
    (d) La media de asistencia diaria.
    (e) Imprimir para cada tabla la diferencia porcentual (positiva ó negativa) entre la asistencia diaria y la media de asistencia a cada sala.

    Como datos de entrada tendremos el número de asistentes para las diferentes salas, para cada uno de los días del congreso. Realizar una función que nos calcule estos datos.

22. Dado un vector X de n elementos reales donde n es impar, diseñar una función para calcular la mediana de ese vector. La mediana se define como el valor mayor que la mitad de los números y menor que la otra mitad.

    Por ejemplo, en el siguiente vector:

$$\begin{array}{ccccccccc} X[1] & X[2] & X[3] & X[4] & X[5] & X[6] & X[7] & X[8] & X[9] \\ 23 & 1 & 12 & 7 & 11 & 5 & -6 & -1 & 35 \end{array}$$

    que está compuesto de nueve elementos, la función debe encontrar que la solución es 7.

23. Representar un número binario en un array de datos de tipo lógico. Realizar las siguientes operaciones sobre números binarios expresados en esta forma:

    - Complemento a 2 de un número binario.
    - Suma y resta binaria.
    - Las operaciones lógicas AND y OR binarias (bit a bit).

24. Escribir un programa que calcule y muestre por pantalla los 20 primeros términos de la sucesión de Fibonacci de orden 4. El programa debe usar **obligatoriamente** una función que calcule y almacene en un vector los k primeros términos de la sucesión de Fibonacci de orden n. Obviamente, k y n serán parámetros del módulo.

    La sucesión de Fibonacci de orden n es una secuencia de números en la que los dos primeros son el 0 y el 1. A partir del tercero, los elementos se calculan como la suma de los n anteriores, si ya hay n elementos disponibles, o la suma de todos los anteriores si hay menos de n elementos disponibles.

    Por ejemplo, la sucesión de Fibonacci de orden 4 sería la siguiente:

    $$0, 1, 1, 2, 4, 8, 15, 29, \ldots$$

25. Para obtener una lista de todos los números primos menores que un determinado número n, se puede utilizar la Criba de Eratóstenes. Ese método consiste en hacer una lista de todos los números desde 2 hasta $n-1$. Tomamos el 2 y tachamos todos los múltiplos de 2. Luego tomamos el siguiente número que se encuentra después de 2 y que esté sin tachar, tachando de nuevo todos sus múltiplos. Repetimos este paso hasta que se acaben los números. Los números que quedaron sin tachar son los que no tienen divisores (salvo el 1 y él mismo) o sea los primos.
    Escribir una función que obtenga los números primos menores que un determinado número n utilizando el método anterior. La función tendrá como entrada un vector de bool y el número n, y marcará en el vector de bool con el valor *true* aquellas casillas cuya posición corresponda con un número primo (menor a n) y con el valor *false* las posiciones que no correspondan con números primos. La posiciones 0 y 1 las marcaremos con *true*.

    Por ejemplo, si $n = 11$ entonces el vector contendrá los valores:

0	1	2	3	4	5	6	7	8	9	10
true	true	true	true	false	true	false	true	false	false	false

26. Realizar una función que acepte una matriz de enteros y devuelva el número de columnas *únicas* de la matriz, es decir, aquellas para las que **NO** existe otra columna en la matriz con los mismos valores.

    Por ejemplo, dada la matriz 4×7

3	1	0	1	3	-4	1
4	5	10	5	4	4	5
5	7	-1	7	5	3	7
7	8	9	8	7	3	8

    la función deberá devolver 2, ya que las únicas columnas no repetidas son la tercera y la sexta.

27. Realizar una función que construya un vector conteniendo todos los elementos menores a 100 del conjunto C definido a continuación:

    (a) El número 1 pertenece a C.
    (b) Si x pertenece a C, entonces $2x + 1$ y $3x + 1$ también pertenecen a C.
    (c) Ningún otro número está en C.

    C tendrá el siguiente aspecto: $\{1, 3, 4, 7, 9, 10, \cdots\}$. El vector resultado debe aparecer ordenado en orden creciente.