Universidad Zaragoza

Práctica 5:
Vectores y matrices

Fundamentos de Informática

Grado en Ingeniería Mecánica

Curso 2017-2018

Universidad de Zaragoza
Escuela de Ingeniería y Arquitectura
Departamento de Informática e Ingeniería de Sistemas
Área de Lenguajes y Sistemas Informáticos

OBJETIVOS

Los objetivos de esta práctica son los siguientes:

- Entender el mecanismo de la agregación y de la indexación para obtener nuevos tipos de datos
- Introducirse en el manejo de estructuras indexadas 1D y 2D (vectores y matrices)
- Introducirse en el manejo de vectores y matrices de registros

1. Operaciones básicas con vectores

Diseña un programa que cuente el número de apariciones de cada letra del alfabeto (independientemente de si es mayúscula o minúscula, "K" es lo mismo que "k") en una frase introducida por teclado, que finaliza en un salto de línea. Ignora la letra "ñ", así como signos de puntuación, cifras y otros caracteres. Puede haber letras mayúsculas y minúsculas. Ten en cuenta solamente las letras. El programa deberá mostrar por pantalla todas las letras que han aparecido al menos una vez por orden alfabético, junto con el número de veces que han aparecido. Deberás utilizar un diseño modular con procedimientos y funciones y paso por parámetros acorde a lo estudiado en la parte teórica de la asignatura. Nota: deberás usar `cin.get(caracter)` para leer del teclado, de lo contrario `cin>>caracter` se saltaría los saltos de línea y tu programa no funcionaría.

Ejemplo de interacción:

```
Introduce una frase:
Dabale arroz a la zorra el abad.
A 8
B 2
D 2
E 2
L 3
O 2
R 4
Z 2
```

Nombre del fichero: apariciones.cpp

2. Operaciones con vectores de registros

Escribe un programa que genere una lista de 5 números complejos con partes real e imaginaria elegidas en un intervalo (a, b) de manera aleatoria. Los límites del intervalo se pedirán al usuario de manera que los valores introducidos verifiquen que a < b. A partir de dicha lista definir procedimientos o funciones para calcular:

- La posición del número complejo de mayor módulo
- La posición del número complejo de menor módulo
- Los números complejos de mayor y menor módulo (definir un único procedimiento)
- La suma de todos los números complejos de la lista

Nota: Si varios números tienen el mismo módulo se elegirá la primera ocurrencia para el cálculo del máximo y del mínimo.

Ejemplo de interacción:

```
Introduce los limites del intervalo (a<b): 1.0 -1.0
Introduce los limites del intervalo (a<b): -1.0 1.0
Numeros complejos generados:
0.2 + 0.3 i, 0.3 + 0.2 i, -0.5 + 0.0 i, 0.1 + 0.4 i, 0.99 + 0.2 i
Numeros complejos de mayor y menor módulo:
0.99 + 0.2 i y 0.2 + 0.3 i
Suma:
1.09 + 1.10 i
```

Nombre del fichero: vectorComplejo.cpp

3. Operaciones con matrices de tipos primitivos

Diseña un programa para trabajar con matrices enteras de tamaño 15 x 10 con valores entre -100 y 100. Desarrollar los siguientes procedimientos y funciones:

```
3.1. void generarMatriz ( vector < vector < int > > & m )

// Genera de manera aleatoria los valores de cada componente de la matriz

3.2. void escribirMatriz ( vector < vector < int > > & m )

// Muestra por la salida estandar el contenido de la matriz m

3.3. vector< vector< int > > trasponer( vector < vector < int > > & m )

// Traspone el contenido de la matriz m

3.4. vector< vector< int > > filtroMax( vector < vector < int > > & m )

// Genera una nueva matriz donde cada componente se calcula a partir del
maximo de los vecinos de dicha componente. Es decir, si (i,j) son los indices
de la componente en cuestion el valor en la nueva matriz se calcula a partir
de max( m(i-1,j-1), m(i-1,j), m(i-1,j+1), m(i,j-1), m(i,j), m(i,j+1), m(i+1,j-
1), m(i+1,j), m(i+1,j+1)). El tratamiento de los bordes de la matriz deberá
ser el adecuado para no acceder a posiciones prohibidas de la memoria.

3.5. int main()

// El algoritmo principal mostrara la funcionalidad de los procedimientos y
funciones desarrollados.
```

Ejemplo de interacción (matriz 2 x 3, valores entre 0 y 10):

```
Matriz generada:
4 5 10
1 2 7
Matriz traspuesta:
4 1
5 2
10 7
Matriz filtrada:
5 10 10
5 10 10
```

Nombre del fichero: **matrizAleatoria.cpp**

4. Operaciones con matrices de registros

Diseña un programa para trabajar con una matriz de fracciones de tamaño 3 x 3 donde el numerador y denominador toman valores entre -10 y 10. Desarrollar los siguientes procedimientos y funciones:

```
3.1. void generarMatriz ( vector < vector < tpFraccion > > & m )

// Genera de manera aleatoria los valores de cada componente de la matriz

3.2. void escribirMatriz ( vector < vector < tpFraccion > > & m )
```

```
// Muestra por la salida estandar el contenido de la matriz m donde cada componente es de la forma a / b

3.3. bool esCorrecta( vector < vector < tpFraccion > > & m )

// Verifica si todos los denominadores de cada componente de la matriz son diferentes de cero

3.4. vector < vector < double > > transformarMatriz( vector< vector < tpFraccion > > & m )

// Transforma la matriz de fracciones a la matriz de componentes reales correspondiente

3.5. int main()

// El algoritmo principal mostrara la funcionalidad de los procedimientos y funciones desarrollados. La función transformarMatriz se invocara solo en caso de que la matriz generada aleatoriamente sea correcta.
```

Ejemplo de interacción:

```
 Matriz generada:
 4/3 5/0 -2/3
 2/7 8/-9 5/4
 0/-3 8/4 -5/4
 No es correcta.
--------------------
 Matriz generada:
 4/3 5/3 -2/3
 2/7 8/-9 5/4
 0/-3 8/4 -5/4
 Si es correcta.
 Matriz transformada:
 1.33 1.66 -0.66
 0.28 -0.88 1.25
0.00 2.00 -1.25
--------------------
```

Nombre del fichero: **matrizFracciones.cpp**

PARA ENTREGAR

Se entregará un fichero **.zip** que contiene el código fuente del trabajo realizado para la práctica tal y como se indicó en la Práctica 1. Para ello se deberá enviar el fichero utilizando el link correspondiente a la Práctica 5 en Moodle. Para evaluación continua se tomará el código fuente de la solución del ejercicio 3 de la práctica (**matrizAleatoria.cpp**). La fecha de entrega será **una semana** después de la finalización de la Práctica 5. Por ejemplo, un alumno que asista a la sesión de 12:00 a 14:00 horas del 8 de mayo de 2018 deberá de entregar la