

LENGUAJES DE PROGRAMACIÓN

INSTRUCCIONES

Por favor, entregue esta primera hoja de enunciado junto con el examen.

Dispone de 2 horas para realizar el examen.

MATERIAL PERMITIDO: Ninguno.

Pregunta 1 (3 puntos)

Indique la veracidad o falsedad de cada una de las afirmaciones siguientes, **explicando detalladamente** en cada caso el motivo de su respuesta.

- A. (0.5 puntos) En el lenguaje Python los bloques de código se delimitan por llaves.
- B. (0.5 puntos) En Java el programador puede especificar completamente el rango de índices del array.
- C. (0.5 puntos) En Java y C++ la declaración de variables puede aparecer en cualquier sitio donde puede aparecer una sentencia.
- D. (0.5 puntos) La verificación de tipos estática se realiza un única vez, durante la traducción del programa a código máquina.
- E. (0.5 puntos) En una pila, el último elemento que se ha añadido es el primero en ser extraído.
- F. (0.5 puntos) Una expresión lambda en C++ es una función anónima, que es creada en tiempo de ejecución.

Pregunta 2 (1 punto)

Escriba la salida por consola producida al ejecutar el programa en C++ escrito a continuación.

```
#include <iostream>
#include <vector>
#include <stdexcept>

enum TError {error1, error2, error3};

int main() {
    int x = 3, y = 8, z = 0;
    std::vector<int> a(3);
    a.push_back(1); a.push_back(2); a.push_back(3);
    bool b1 = !( (x < 3) || (y > 7) );
    for (int i = 1; i < 5; i++) {
        try {
            switch (i) {
                case 1: { throw x; }
                case 2: { TError mte1 = error1; throw mte1; }
                case 3: { TError mte1 = error3; throw mte1; }
                case 4: { std::cout << "b1 = " << std::boolalpha
                        << b1 << std::endl;
                        TError mte1 = error2;
                        throw mte1; }
            }
        } catch (int exc) {
            std::cout << x << std::endl;
        } catch (TError exc) {
            std::cout << "Excepcion TError: "
                << exc << std::endl;
        } catch (...) {
            std::cout << "Excepcion otro tipo" << std::endl;
        }
    }
    try {
        std::cout << a.at(10) << std::endl;
    } catch (const std::out_of_range) {
        std::cout << "ERROR";
        return 1 ;
    }
    std::cout << "Programa finalizado." << std::endl;
    return 0;
}
```

Pregunta 3 (1 punto)

Escriba la salida por consola producida al ejecutar el programa en C++ escrito a continuación.

```
#include <iostream>
#include <list>
#include <algorithm>

int main() {
    std::list<double> lista01 { -1.1, 2.8, -1.3, 5.6,
                                2e3, 10e4, 9 };
    std::list<double> lista02 { 4.4, 3.8, 7.2, -6.33,
                                1.2e3, 10.8, 7.4 };
    std::list<double> listaD(lista01.size());

    std::list<double>::iterator pEndD =
        std::transform( lista01.begin(), lista01.end(),
                        lista02.begin(), listaD.begin(),
                        [](double x1, double x2) -> double {
                            return (x1 > x2) ? -x1 : x2;
                        }
        );

    std::cout << "listaD:";
    for ( std::list<double>::iterator p = listaD.begin();
          p != pEndD; p++ )
        std::cout << " " << (*p);

    return 0;
}
```

Pregunta 4 (2.5 puntos)

El cifrado BIFID es un método de cifrado por sustitución que se desarrolló a finales de siglo XIX. El método emplea la siguiente matriz de letras (obsérvese que la letra J no está presente en la matriz).

$$\begin{pmatrix} A & B & C & D & E \\ F & G & H & I & K \\ L & M & N & O & P \\ Q & R & S & T & U \\ V & W & X & Y & Z \end{pmatrix}$$

Los elementos de la matriz se designan mediante su número de fila y columna, las cuales comienzan a numerarse por cero. Por ejemplo, la letra R es el elemento (3,1) de la matriz. La codificación de un mensaje se realiza de la forma siguiente:

1. Se reemplaza en el mensaje, tantas veces como aparezca, la letra J por I.
2. Se obtiene el número de fila y columna en la matriz de cada una de las letras del mensaje a cifrar. Los espacios en blanco del mensaje son ignorados. Por ejemplo, si se desea cifrar el mensaje REUNION LUNES, el número de fila y columna en la matriz de cada uno de los caracteres es:

R (3, 1) E (0, 4) U (3, 4) N (2, 2) I (1, 3) O (2, 3) N (2, 2)
L (2, 0) U (3, 4) N (2, 2) E (0, 4) S (3, 2)

3. Se escriben primeramente todos los números de fila de las letras del mensaje y a continuación todos los números de columna. En el ejemplo anterior, obtenemos la secuencia siguiente:

3, 0, 3, 2, 1, 2, 2, 2, 3, 2, 0, 3, 1, 4, 4, 2, 3, 3, 2, 0, 4, 2, 4, 2

4. Se agrupan de dos en dos los números de la secuencia obtenida. En el ejemplo anterior:

(3, 0) (3, 2) (1, 2) (2, 2) (3, 2), (0, 3) (1, 4) (4, 2) (3, 3)
(2, 0) (4, 2) (4, 2)

El primer número de cada pareja se interpreta como número de fila y el segundo como número de columna, de manera que cada pareja de números designa un elemento de la matriz: la letra cifrada. En el ejemplo, es obtiene:

Q (3, 0) S (3, 2) H (1, 2) N (2, 2) S (3, 2) D (0, 3) K (1, 4)
X (4, 2) T (3, 3) L (2, 0) X (4, 2) X (4, 2)

Por lo que el mensaje cifrado es: QSHNSDKXTLXX

Escriba un programa en C++ que realice las acciones siguientes:

1. Solicitar al usuario que introduzca un mensaje por consola. Almacenar el mensaje en una variable *string* llamada `mensaje`.
2. Si el mensaje contiene algún caracter no válido (es decir, diferente de las letras de la matriz, la letra J y el espacio en blanco), mostrar un mensaje en la consola indicándolo y terminar. En caso contrario, obtener el mensaje cifrado BIFID y escribirlo en la consola.
3. Terminar.

Pregunta 5 (2.5 puntos)

En un fichero de texto llamado *tarea.txt* está almacenada la información de los paquetes que debe entregar un repartidor durante su jornada laboral. Cada entrega está descrita en una línea del fichero. En la primera columna del fichero está escrito el identificador de la entrega, el cual es una palabra compuesta de letras mayúsculas y/o números, que designa de manera unívoca la entrega. En la segunda y tercera columnas están las coordenadas cartesianas X e Y del punto donde debe entregarse el paquete. Estas coordenadas representan distancias medidas en kilómetros.

Se muestra a continuación un ejemplo del contenido del fichero *tarea.txt*. Las líneas del fichero no están ordenadas siguiendo ningún criterio de ordenación relativa.

J67A6	0.6	-1.2
WU1D	2.4	3.6
TYTT5	-0.9	-2.4
NJYR9	3.2	2.5
F8N	-6.4	3.3
Z35D	5.1	-4.3

El punto inicial y final de la ruta diaria de reparto es siempre el mismo: la base logística, que está ubicada en el origen de coordenadas (X=0, Y=0).

Escriba el código C++ indicado a continuación.

5.1 (0.5 puntos) Escriba una función en C++ con el prototipo siguiente:

```
std::list<Entrega> leeFich()  
    throw (std::invalid_argument);
```

La función debe abrir el fichero de texto *tarea.txt* para lectura. Si se produce error, debe lanzar una excepción. En caso contrario, la función debe almacenar el contenido del fichero en una lista de elementos del tipo estructura `Entrega`, cuya declaración se muestra a continuación, y devolver la lista.

```
struct Entrega {  
    std::string id;  
    double x, y;  
};
```

5.2 (0.5 puntos) Escriba una función en C++ con el prototipo siguiente:

```
double distEntregas(std::string ID1, std::string ID2,  
                   std::list<Entrega> &lEntrega)  
    throw (std::invalid_argument);
```

que devuelva la distancia en línea recta, expresada en km, entre las posiciones de las entregas cuyos identificadores coinciden con los dos primeros argumentos. El tercer argumento de la función es la lista donde se ha almacenado el contenido del fichero *tarea.txt*. Si alguno de los identificadores no se encuentra en la lista, la función debe lanzar una excepción.

5.3 (0.5 puntos) Escriba una función en C++ con el prototipo siguiente:

```
std::map<std::string, double> distBase(  
    std::list<Entrega> &lEntrega);
```

que devuelva un mapa que contenga como clave los ID de las entregas de la lista pasada como argumento, y como valor la distancia en línea recta, expresada en km, entre cada uno de los puntos de entrega y la base logística (situada en el origen de coordenadas).

5.4 (1 punto) Escriba un programa principal en C++ que realice las acciones siguientes:

1. Solicitar al usuario, mediante la escritura de un mensaje en la consola, que introduzca por consola los identificadores de dos entregas. Leer los valores introducidos por el usuario, almacenándolos en sendos *strings* llamados ID1 e ID2.
2. Abrir el fichero de texto *tarea.txt* y almacenar su contenido en una lista de elementos del tipo estructura *Entrega* llamada *lEntregas*. Para ello, invocar la función definida en el Apartado 5.1. Si la función lanza una excepción, capturarla y escribir en la consola un mensaje indicando que no ha sido posible abrir el fichero y terminar.
3. Invocando la función definida en el Apartado 5.2, pasando como argumento la lista *lEntregas*, obtener la distancia en línea recta, expresada en km, entre los dos puntos de entrega indicados por el usuario y escribirla en la consola en formato fijo con un dígito decimal. Si la función lanza una excepción, escribir un mensaje en la consola indicando que al menos uno de los puntos de entrega no está en la lista y terminar.
4. Invocando la función definida en el Apartado 5.3, obtener un mapa con la distancia a la base de los puntos de entrega de la lista *lEntregas*. Buscar en este mapa la distancia a la base de cada uno de los dos puntos de entrega introducidos por el usuario y escribir en la consola estas distancias, expresadas en km, en formato fijo con un dígito decimal.
5. Terminar.