

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) Pascal no permite inicializar una variable al declararla.
- B. (0.5 puntos) Java soporta sólo un tipo de dato entero, que tiene 128 bits.
- C. (0.5 puntos) Para poder acceder a todos los elementos de un array los compiladores han de almacenar la dirección de memoria de cada uno de sus elementos.
- D. (0.5 puntos) La recursividad de cola es un tipo de recursividad lineal.
- E. (0.5 puntos) En C y C++ la expresión de control del bucle **for** es opcional.
- F. (0.5 puntos) El algoritmo `transform` aplica una función especificada a cada uno de los elementos de la secuencia origen, almacenando el resultado en la misma secuencia origen.

Pregunta 2 (1 punto)

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

```
#include <iostream>

int main() {
    int *p1, *p2;
    int j = 7;
    int k = 21;
    p1 = new int;
    *p1 = j;
    std::cout << *p1 << std::endl;
    p2 = p1;
    std::cout << *p1 << std::endl;
    std::cout << *p2 << std::endl;
    *p2 = k;
    std::cout << *p1 << std::endl;
    std::cout << *p2 << std::endl;
    p1 = new int;
    std::cout << *p2 << std::endl;
    *p1 = 8;
    std::cout << *p1 << std::endl;
    std::cout << *p2 << 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 <vector>
#include <algorithm>

int main()
{
    std::vector<int> vec(4,2);
    vec.push_back(0);  vec.push_back(1);  vec.push_back(0);
    vec.push_back(-5); vec.push_back(-4); vec.push_back(-6);

    for (unsigned int i=0; i<vec.size(); i++)
        std::cout << vec.at(i) << " ";
    std::cout<<std::endl;
    for (int val=0; val<3; val++)
        std::cout <<
            count(vec.begin(),vec.end(),val) << std::endl;
    std::cout << "\n";
    for (int val=0; val<3; val++)
        std::cout <<
            count(vec.begin()+2,vec.end()-1,val) << std::endl;
    return 0;
}
```

Pregunta 4 (2.5 puntos)

La sucesión de Thue-Morse es una sucesión de números binarios que comienza siempre con el primer valor 0. La sucesión puede obtenerse de manera iterativa, comenzando por el valor cero y concatenando el complemento Booleano de la sucesión existente hasta el momento. A continuación se muestra la sucesión s_4 , obtenida mediante 4 iteraciones del método anteriormente descrito, donde $C(s)$ representa el complemento Booleano de s y el operador $+$ representa concatenación.

$$\begin{aligned}s_1 &= 0 \\s_2 &= s_1 + C(s_1) = 01 \\s_3 &= s_2 + C(s_2) = 0110 \\s_4 &= s_3 + C(s_3) = 01101001\end{aligned}$$

4.1 (1.5 puntos) Escriba una función en C++ cuyo prototipo sea:

```
std::vector<bool> sucThueMorse(int N)
    throw ( std::invalid_argument );
```

La funcionalidad debe ser la siguiente:

- Si N es menor que uno, la función lanza una excepción.
- En caso contrario, aplica el método anteriormente descrito para obtener la sucesión s_N y la devuelve como un vector de componentes Booleanos.

4.2 (1 punto) Escriba en C++ un programa principal que realice las acciones siguientes:

1. Escriba un mensaje en la consola solicitando al usuario que éste introduzca por consola el valor de N .
2. El valor introducido por el usuario debe almacenarse en una variable entera llamada N .
3. Invocar la función `sucThueMorse` definida anteriormente, pasándole N como argumento.
4. Si la función lanza una excepción, mostrar un mensaje en la consola indicándolo. En caso contrario, escribir en la consola el contenido del vector devuelto por la función.
5. Terminar.

Pregunta 5 (2.5 puntos)

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

1. Abrir para lectura un fichero de texto llamado “datos.txt”. Si se produce error, terminar.
2. Leer palabra a palabra el fichero.
3. Calcular el número de palabras que son iguales a cada una de las palabras siguientes: 0, 01, 0110, 01101001.

La estructura de datos que debe emplearse para ello es un mapa. La clave debe ser el número de caracteres de la palabra: 1, 2, 4 u 8. El valor para la clave 1, 2, 4 y 8 debe ser el número de palabras leídas del fichero iguales a 0, 01, 0110 y 01101001, respectivamente.

4. Mostrar en la consola el contenido del mapa.
5. Terminar.

A continuación se muestra un ejemplo. Supongamos que el contenido del fichero es:

```
1  01  0  01101001 01100110 01 01
```

El mensaje escrito en consola por el programa debería ser:

```
1, 3, 0, 1
```

indicando que en el fichero aparece la palabra 0 una vez, tres veces la palabra 01, ninguna vez la palabra 0110 y una vez la palabra 01101001.