

# LENGUAJES DE PROGRAMACIÓN

## Solución al examen de Junio 2018, Segunda Semana

### 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 FORTRAN I no se declara el tipo de la variable.
- B. (0.5 puntos) En C++, el tipo de acceso a los elementos del tipo de dato `std::list` es aleatorio y hacia delante.
- C. (0.5 puntos) El lenguaje Java no presenta el problema de las variables dinámicas perdidas.
- D. (0.5 puntos) Se define la altura de un nodo en un árbol como la longitud del camino más corto desde el nodo hasta una hoja.
- E. (0.5 puntos) Al implementar la lista empleando un array, resulta eficiente almacenar los elementos en los primeros componentes del array.
- F. (0.5 puntos) La función `list.end()` devuelve un iterador a la posición del último elemento de la lista.

### Solución a la Pregunta 1

- A Verdadero Véase la página 43 del texto base.
- B Falso Véase la página 455 del texto base.
- C Verdadero Véase la página 127 del texto base.
- D Falso Véase la página 438 del texto base.
- E Falso Véanse las páginas 443 y 447 del texto base.
- F Falso Véase la página 456 del texto base.

**PREGUNTA 2** (1 punto)

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

```
#include <iostream>
#include <iomanip>

int main() {
    double i = 112.2379;
    double a[4] = {22.123, 61.551, 3.2343, 390.2323};
    double *p;
    p = &i;
    std::cout << std::scientific << std::setprecision(3)
              << *p << std::endl;
    std::cout << i << std::endl;
    *p = 137.98;
    std::cout << std::scientific << std::setprecision(4)
              << *p << std::endl;
    std::cout << i << std::endl;
    p = a;
    p++;
    *p = 29.242;
    p++;
    std::cout << *p << std::endl;
    std::cout << i << std::endl;
    std::cout << a[1] << std::endl;
    return 0;
}
```

**Solución a la Pregunta 2**

```
1.122e+002
1.122e+002
1.3798e+002
1.3798e+002
3.2343e+000
1.3798e+002
2.9242e+001
```

**PREGUNTA 3** (1 punto)

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

```
#include <iostream>
#include <string>
#include <vector>
#include <algorithm>

void ivector(std::vector<int> &v) {
    for (unsigned int i=0; i<v.size()-1; i++)
        std::cout << v[i] << ", ";
    std::cout << v[v.size()-1] << std::endl;
    return;
}

int main() {
    std::vector<int> v;
    for (int i=0; i<10; i++)
        v.push_back(2*i);
    ivector(v);
    std::vector<int>::iterator pBegin = v.begin()+2;
    std::vector<int>::iterator pEnd = v.begin()+7;
    std::cout << "(*pBegin): " << (*pBegin)
                << "\t(*pEnd): " << (*pEnd) << std::endl;
    v.push_back(100);
    ivector(v);
    reverse(pBegin,pEnd);
    ivector(v);
    std::cout << "(*pBegin): " << (*pBegin)
                << "\t(*pEnd): " << (*pEnd) << std::endl;
    return 0;
}
```

**Solución a la Pregunta 3**

```
0,2,4,6,8,10,12,14,16,18
(*pBegin): 4 (*pEnd): 14
0,2,4,6,8,10,12,14,16,18,100
0,2,12,10,8,6,4,14,16,18,100
(*pBegin): 12 (*pEnd): 14
```

**PREGUNTA 4** (2 puntos)

El *método de Newton* es un algoritmo muy eficiente para resolver  $f(x) = 0$ . Partiendo de un valor inicial  $x_0$ , el método de Newton obtiene  $x_1, x_2, \dots$ , mediante la fórmula:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} \quad \text{con } k = 0, 1, \dots$$

Escriba un programa en C++ que aplique el método de Newton para resolver:

$$f(x) = x^2 - 2 = 0$$

Puesto que la derivada es:

$$f'(x) = 2 \cdot x$$

obtenemos:

$$x_{k+1} = x_k - \frac{f(x_k)}{f'(x_k)} = x_k - \frac{x_k^2 - 2}{2 \cdot x_k} \quad \text{con } k = 0, 1, \dots$$

El programa debe realizar las acciones siguientes:

1. Solicitar al usuario que introduzca por consola el valor inicial ( $x_0$ ) y el número de iteraciones del algoritmo ( $N$ ). Leer los valores introducidos.
2. Comprobar que el número de iteraciones es mayor que cero. Si no se satisface esta condición, terminar.
3. Escribir en la consola, para  $k = 0, \dots, N$ , los valores  $x_k$  obtenidos de aplicar el método de Newton y los correspondientes valores  $f(x_k)$ . Ambos valores deben mostrarse en formato científico, con 10 dígitos de precisión.
4. Terminar.

## Solución a la Pregunta 4

```
#include <iostream>
#include <iomanip>
#include <cmath>

int main()
{
    // Entrada por consola del valor inicial
    std::cout << "Valor inicial: ";
    double x_k;
    std::cin >> x_k;
    // Entrada por consola del número de iteraciones
    std::cout << "Numero de iteraciones: ";
    int N;
    std::cin >> N;
    // Comprobar que N es mayor que cero
    if (N<=0) return 0;
    // Método de Newton
    for (int k=0; k<=N; k++) {
        //f(x_k)
        double f_k = std::pow(x_k,2) - 2;
        // Salida por consola
        std::cout << k << "\t" <<
            std::scientific << std::setprecision(10) <<
            x_k << "\t" << f_k << "\t" << std::endl;
        //f'(x_k)
        double derf_k = 2*x_k;
        // Iteración del método
        x_k = x_k - f_k/derf_k;
    }
    return 0;
}
```

**PREGUNTA 5 (3 puntos)**

En las técnicas criptográficas de *cifrado por sustitución*, el mensaje cifrado se obtiene sustituyendo cada letra del mensaje original por otra letra, empleando para ello un determinado algoritmo.

Una de estas técnicas de cifrado por sustitución consiste en establecer la correspondencia entre el alfabeto original y el alfabeto cifrado en base a una clave que sólo conozcan el emisor y el receptor del mensaje. A continuación se describe un método sencillo.

Supongamos que la clave escogida para realizar el cifrado es:

JULIUS CAESAR

El primer paso consiste en eliminar de la clave los espacios en blanco y los caracteres repetidos. Obtenemos la clave manipulada:

JULISCAER

Empleamos la clave manipulada como comienzo del alfabeto cifrado. El resto del alfabeto cifrado consiste en las letras restantes del alfabeto, escritas en su orden correcto comenzando por la última letra de la clave manipulada.

Original: a b c d e f g h i j k l m n o p q r s t u v w x y z  
Cifrado: J U L I S C A E R T V W X Y Z B D F G H K M N O P Q

Obsérvese que la letra ñ no forma parte de los alfabetos.

Obsérvese también que hemos aplicado el criterio de escribir el alfabeto original empleando letras minúsculas y el alfabeto cifrado empleando letras mayúsculas.

Escriba el código C++ indicado a continuación.

**5.1 (1.5 puntos)** Escriba una función en C++ con el prototipo siguiente:

```
std::string manipulaClave(std::string claveCifrado)
    throw (std::invalid_argument);
```

El parámetro de la función es la clave escogida por el usuario.

La función debe comprobar que el string pasado como parámetro está compuesto únicamente por una o más letras mayúsculas del conjunto

{A, B, C, D, E, F, G, H, I, J, K, L, M, N, O, P, Q, R, S, T, U, V, W, X, Y, Z}

y por cero, uno o más espacios en blanco. Si el string pasado como argumento no supera esta comprobación, la función debe lanzar una excepción.

El string devuelto por la función es la clave manipulada. Es decir, el resultado obtenido al eliminar los espacios en blanco y los caracteres repetidos del string pasado como parámetro. Por ejemplo, si se pasa como parámetro "JULIUS CAESAR", el valor devuelto debe ser "JULISCAER".

**5.2** (1 punto) Escriba una función en C++ con el prototipo siguiente:

```
std::string alfabetoCifrado(std::string claveManipulada);
```

El string pasado como argumento es la clave modificada. El valor devuelto por la función es el alfabeto cifrado.

Por ejemplo, si se pasa como parámetro "JULISCAER", la función debe devolver

"JULISCAERTVWXYZBDFGHKMNOPQ".

**5.3** (0.5 puntos) Escriba un programa en C++ que realice las acciones siguientes:

1. Solicitar al usuario, mediante un mensaje en la consola, que introduzca por consola la clave.
2. Leer la clave de la consola, almacenándola en un string. Tenga en cuenta que la clave puede contener espacios en blanco. Por ello, el programa debe leer una línea de texto.
3. Empleando esta clave, e invocando las dos funciones definidas en los Apartados 5.1 y 5.2, obtener el alfabeto cifrado, mostrándolo a continuación en la consola.

Si la función `manipulaClave` lanza una excepción, mostrar un mensaje en la consola indicándolo y terminar.

4. Terminar.

## Solución a la Pregunta 5

```

#include <iostream>
#include <string>
#include <sstream>
#include <stdexcept>

const std::string alf_M = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";

std::string manipulaClave(std::string claveCifrado)
    throw (std::invalid_argument){
    // Comprobación del parámetro
    bool claveSinLetras = true;
    for (unsigned int i=0; i<claveCifrado.length(); i++) {
        if (claveCifrado[i] != ' '){
            int p = alf_M.find(claveCifrado[i]);
            if (p < 0 || p >= alf_M.length())
                throw std::invalid_argument("Caracter incorrecto en clave");
            claveSinLetras = false;
        }
    }
    if (claveCifrado.length() == 0 || claveSinLetras )
        throw std::invalid_argument("La clave no contiene letras");
    // Eliminar blancos y caracteres repetidos
    std::stringstream ss;
    for (unsigned int i=0; i<claveCifrado.length(); i++) {
        if (claveCifrado[i] != ' '){
            int p = ss.str().find(claveCifrado[i]);
            if (p<0 || p>=ss.str().length())
                ss << claveCifrado[i];
        }
    }
    return ss.str();
}

```

```

std::string alfabetoCifrado(std::string claveManipulada){
    if (claveManipulada.length() == 0)
        return alf_M;
    // Rotar el alfabeto, de modo que la letra siguiente a
    // la última letra de la clave sea la primera del alfabeto rotado
    int pUL = alf_M.find(claveManipulada[claveManipulada.length()-1]);
    std::string alf_rot;
    if (pUL == alf_M.length()-1){
        alf_rot = alf_M;
    } else {
        alf_rot = alf_M.substr(pUL+1,alf_M.length()-pUL-1)
            + alf_M.substr(0,pUL+1);
    }
    // Alfabeto cifrado
    std::stringstream ss;
    ss << claveManipulada;
    for (unsigned int i=0; i<alf_rot.length(); i++){
        int p = claveManipulada.find(alf_rot[i]);
        if (p<0 || p>=claveManipulada.length())
            ss << alf_rot[i];
    }
    return ss.str();
}

int main(){
    // Entrada por consola de la clave
    std::string clave;
    std::cout << "Introduzca clave:" << std::endl;
    getline(std::cin,clave);
    // Manipulación de la clave
    // Obtención y escritura en consola del alfabeto cifrado
    try {
        std::cout << "Alfabeto cifrado:\n"
            << alfabetoCifrado(manipulaClave(clave))
            << std::endl;
    } catch (std::invalid_argument exc) {
        std::cout << exc.what() << std::endl;
        return 0;
    }
    return 0;
}

```