

# LENGUAJES DE PROGRAMACIÓN

## Solución al Ejercicio de Autocomprobación 2

### PREGUNTA 1 (1 punto)

Señale razonadamente la veracidad o falsedad de las afirmaciones siguientes:

- A. En el contador de programa de la máquina de von Neumann se almacena el siguiente dato con el que va a operar la unidad aritmética.
- B. ALGOL 60 introdujo el concepto de bloque de código.
- C. El programa que enlaza el código objeto con los programas del sistema operativo y con otro código objeto se denomina intérprete.
- D. El preprocesador se encarga de procesar un programa inmediatamente antes de que éste sea compilado.

### Solución a la Pregunta 1

La opción A es falsa, ya que el contador de programa almacena la dirección de memoria en la cual se encuentra la siguiente instrucción a ejecutar del programa.

La opción B es verdadera.

La opción C es falsa, ya que este programa se llama enlazador o *linker*.

La opción D es verdadera.

**PREGUNTA 2** (1 punto)

Señale razonadamente la veracidad o falsedad de las afirmaciones siguientes:

- A. En el lenguaje C++, si el programador no inicializa una variable o un array de un tipo básico, la implementación siempre la inicializa a su valor por defecto.
- B. El operador **delete** de C++ sólo puede ser aplicado a un puntero que apunta a una variable en memoria dinámica.
- C. La sentencia

$$x = i++;$$

es equivalente a estas dos sentencias:

$$x = i; \quad i = i + 1;$$

- D. Los operadores relacionales siempre tienen menor precedencia que los aritméticos.

**Solución a la Pregunta 2**

La opción A es falsa, ya que esto sólo ocurre si se trata de una variable global. Si es una variable o array local a algún bloque de código, la implementación no la inicializa.

La opción B es verdadera. Si se aplica sobre un puntero que apunta a otro tipo de variable, entonces se produce un error en la ejecución del programa.

La opción C es verdadera.

La opción D es verdadera.

**PREGUNTA 3** (1 punto)

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

```
#include <iostream>
int main()
{
    std::cout << "\tValor" << std::endl;
    int *p1, *p2;
    p1 = new int;
    *p1 = 5;
    std::cout << "\t*p1: " << *p1 << std::endl;
    p2 = p1;
    std::cout << "\t*p1: " << *p1 << std::endl;
    std::cout << "\t*p2: " << *p2 << std::endl;
    *p2 = 2;
    std::cout << "\t*p1: " << *p1 << std::endl;
    std::cout << "\t*p2: " << *p2 << std::endl;
    p2 = new int;
    std::cout << "\t*p1: " << *p1 << std::endl;
    std::cout << "\t*p2: " << *p2 << std::endl;
    *p1 = 8;
    std::cout << "\t*p1: " << *p1 << std::endl;
    std::cout << "\t*p2: " << *p2 << std::endl;
    return 0;
}
```

**Solución a la Pregunta 3**

El programa produce la salida por consola mostrada a continuación. El valor apuntado por `p2` en las dos últimas salidas por consola es un valor "basura", que puede ser diferente de 4063608.

```
Valor
*p1: 5
*p1: 5
*p2: 5
*p1: 2
*p2: 2
*p1: 2
*p2: 4063608
*p1: 8
*p2: 4063608
```

**PREGUNTA 4** (1 punto)

Señale razonadamente la veracidad o falsedad de las afirmaciones siguientes:

- A. En FORTRAN 90 no existen bucles controlados mediante expresión Booleana.
- B. La sentencia **continue** de Java hace que finalice la ejecución del bucle, pasando el control a la sentencia siguiente al bucle.
- C. Los lenguajes C y FORTRAN no facilitan la gestión de las excepciones.
- D. El tipo de la excepción en C++ puede ser un tipo básico del lenguaje, uno de los tipos declarados en la librería estándar de C++ o un tipo definido por el usuario.

**Solución a la Pregunta 4**

La opción A verdadera.

La opción B es falsa, ya que la sentencia **continue** hace que el control pase al comienzo del bucle.

Las opciones C y D son verdaderas.

**PREGUNTA 5** (1 punto)

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

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

std::string listaInt2string(std::list<int> &lst) {
    std::stringstream ss;
    if (lst.empty()) {
        ss << "(lista vacia)";
    } else {
        std::list<int>::iterator p = lst.begin();
        ss << "(" << lst.size() << " elementos):\t";
        while (p!=lst.end()) {
            ss << *p << " ";
            p++;
        } //FIN WHILE
    } //FIN IF
    return ss.str();
}

int main()
{
    std::list<int> lista1, lista2;
    for (int i=0; i<6; ++i) {
        lista1.push_back(i);
        lista2.push_front(i);
    }
    std::cout << "lista1 -1 " << listaInt2string(lista1) << std::endl;
    std::cout << "lista2 -1 " << listaInt2string(lista2) << std::endl;
    lista2.splice(find(lista2.begin()), lista2.end(), 3, lista1);
    std::cout << "lista1 -2 " << listaInt2string(lista1) << std::endl;
    std::cout << "lista2 -2 " << listaInt2string(lista2) << std::endl;
    lista2.splice(lista2.end(), lista2, lista2.begin());
    std::cout << "lista1 -3 " << listaInt2string(lista1) << std::endl;
    std::cout << "lista2 -3 " << listaInt2string(lista2) << std::endl;
    lista2.sort();
    lista1 = lista2;
    std::cout << "lista1 -4 " << listaInt2string(lista1) << std::endl;
    std::cout << "lista2 -4 " << listaInt2string(lista2) << std::endl;
    lista1.merge(lista2);
    std::cout << "lista1 -5 " << listaInt2string(lista1) << std::endl;
    std::cout << "lista2 -5 " << listaInt2string(lista2) << std::endl;
}
```

## Solución a la Pregunta 5

La salida por consola obtenida al ejecutar el programa se muestra a continuación.

```
lista1 -1 (6 elementos): 0 1 2 3 4 5
lista2 -1 (6 elementos): 5 4 3 2 1 0
lista1 -2 (lista vacia)
lista2 -2 (12 elementos): 5 4 0 1 2 3 4 5 3 2 1 0
lista1 -3 (lista vacia)
lista2 -3 (12 elementos): 4 0 1 2 3 4 5 3 2 1 0 5
lista1 -4 (12 elementos): 0 0 1 1 2 2 3 3 4 4 5 5
lista2 -4 (12 elementos): 0 0 1 1 2 2 3 3 4 4 5 5
lista1 -5 (24 elementos): 0 0 0 0 1 1 1 1 2 2 2 2 3 3 3 3
                        4 4 4 4 5 5 5 5
lista2 -5 (lista vacia)
```

**PREGUNTA 6 (2 puntos)**

Programe una función recursiva, denominada `mcd`, con dos argumentos enteros y que devuelva un valor entero: el máximo común divisor de los dos argumentos. La declaración de la función es:

```
int mcd(int, int);
```

El cálculo debe realizarse aplicando el algoritmo de Euclides, que se describe a continuación.

Supongamos que se desea calcular el máximo común divisor de dos números enteros, a los que llamaremos  $a$  y  $b$ , satisfaciéndose que  $a \geq b$  y  $b \neq 0$ . Es posible demostrar que:

$$\text{mcd}(a, b) = \text{mcd}(b, a \bmod b)$$

donde  $a \bmod b$  es el resto de la división entera de  $a$  por  $b$ .

Asimismo, debe tenerse en cuenta que:

$$\text{mcd}(a, 0) = a$$

Por ejemplo:  $\text{mcd}(30, 12) = \text{mcd}(12, 6) = \text{mcd}(6, 0) = 6$

**Solución a la Pregunta 6**

La solución del ejercicio se muestra en Código 1.1.

```
int mcd (int a, int b) {
    if (a < b) return mcd(b,a);
    if (b == 0) return a;
    return mcd(b, a % b);
}
```

**Código 1.1:** Función que calcula el máximo común divisor.

**PREGUNTA 7** (3 puntos)

Programa una función llamada `monteCarlo` que calcule, aplicando el método de Monte Carlo, la integral definida de cualquier función de una variable y el error en la estimación de la integral. La declaración de la función es:

```
double monteCarlo(double (*f)(double),
                  double a, double b, int N,
                  double& errest);
```

donde `f` es la función a integrar, `a` y `b` son, respectivamente, los límites inferior y superior del intervalo de integración, `N` es la cantidad de números aleatorios usados en el cálculo de la integral y `errest` es el error en la estimación de la integral.

El método que debe emplearse para estimar el valor de la integral y el error cometido es el siguiente. Supongamos que se generan  $N$  observaciones independientes,  $x_1, \dots, x_N$ , de una variable aleatoria que está uniformemente distribuida en el intervalo  $[a, b]$ . Puede estimarse el valor de la integral de la función  $f(x)$  en el intervalo  $[a, b]$  de la forma siguiente:

$$\int_a^b f(x) dx = (b - a) \cdot |f|$$

donde

$$|f| = \frac{1}{N} \cdot \sum_{i=1}^N f(x_i)$$

El error en la estimación de la integral se calcula de la expresión siguiente:

$$errest = (b - a) \cdot \sqrt{\frac{|f^2| - |f|^2}{N}}$$

donde

$$|f^2| = \frac{1}{N} \cdot \sum_{i=1}^N f^2(x_i)$$

## Solución a la Pregunta 7

La solución del ejercicio se muestra en el Código 1.2.

```

#include <cstdlib>
#include <iostream>
#include <cmath>

double monteCarlo(double (*f)(double), double a, double b, int N, double& errest)
{
    double sum = 0.0, sumsq = 0.0;
    double x, fx, val;
    for (int i = 0; i < N; i++) {
        x = a + double(rand())/(RAND_MAX+1) * (b - a);
        fx = f(x);
        sum += fx;
        sumsq += fx * fx;
    }
    val = (b-a)*sum/N;
    sum = sum/N;
    sumsq = sumsq/N;
    errest = (b-a)*std::sqrt((sumsq - sum*sum)/N);
    return val;
}

```

**Código 1.2:** Función que estima, aplicando el método de Monte Carlo, la integral definida de una función de una variable y el error cometido.