LENGUAJES DE PROGRAMACIÓN

Solución al examen de Junio 2022, 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) La máquina de Von Neumann soporta las operaciones de suma, resta, multiplicación, división y del cálculo de valor absoluto de un número.

B. (0.5 puntos) La variable del bucle de Pascal es visible dentro del cuerpo del bucle pero no se permite modificar su valor mediante asignaciones dentro del cuerpo del bucle.

C. (0.5 puntos) El rango de valores que puede tomar cada tipo de dato físico en C++ está fijado por las reglas del lenguaje.

D. (0.5 puntos) Cualquier nodo de un árbol tiene cero o un nodo hijo.

E. (0.5 puntos) La siguiente sentencia en lenguaje C++

```
int * fun(int x, double y);
```

es una declaración de un puntero a la función `fun`, la cual devuelve un valor de tipo `int`.

F. (0.5 puntos) La función sobre el flujo de entrada `std::cin.peek()` devuelve el último carácter del flujo de entrada.

Solución a la Pregunta 1

A	Verdadero	Véase la página 30 del texto base.
B	Verdadero	Véase la página 258 del texto base.
C	Falso	Véase la página 132 del texto base.
D	Falso	Véase la página 470 del texto base.
E	Falso	Véase la página 405 del texto base.
F	Falso	Véase la página 323 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 <stdexcept>
#include <string>

bool funciond(std::string s) throw (std::invalid_argument) {
    bool par = false;
    if (s.length() < 4 || s.length() > 4)
        throw std::invalid_argument("ERROR LONGITUD");
    else {
        for (int i = 0; i < 4; i++) {
            if ((s[i] != '0') && (s[i] != '1'))
                throw std::invalid_argument("ERROR VALOR");
        }
    }
    if (s.at(3) == '0')
        par = true;
    return par;
}

int main() {
    std::string s[] = {
        "1001",  "abs",  "ab0z", "1010",
        "10110", "1101", "1001" };
    for (int i = 0; i < 6; i++) {
        try {
            std::cout << (s[i]) << "  ";
            std::cout << funciond(s[i]) << std::endl;
        }
        catch (const std::invalid_argument& exc) {
            std::cout << exc.what() << std::endl;
        }
    }
    return 0;
}
```

Solución a la Pregunta 2

La salida por consola producida al ejecutar el programa es:

```
1001  0
abs  ERROR LONGITUD
ab0z  ERROR VALOR
1010  1
10110  ERROR LONGITUD
1101  0
```

PREGUNTA 3 (1 punto)

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

```
#include <iostream>

int main() {
    int* p;
    int a[] = { 4, 2, 5, 1 };
    p = &a[1];
    std::cout << p[1] << std::endl;
    std::cout << p[2] << std::endl;
    p = a;
    *p = 10;
    p = p + 1;
    std::cout << a[0]     << std::endl;
    std::cout << a[2]     << std::endl;
    std::cout << *p       << std::endl;
    std::cout << *(p - 1) << std::endl;
    return 0;
}
```

Solución a la Pregunta 3

La salida por consola producida al ejecutar el programa es:

```
5
1
10
5
2
10
```

PREGUNTA 4 (2 puntos)

El ayuntamiento del pequeño pueblo de Greenville, Carolina del Norte, lleva guardando desde hace más de 150 años el registro medio mensual de las temperaturas. Dichas medias mensuales se encuentran escritas por orden cronológico (de más antiguo, a más moderno), formando una única columna de números reales, en un fichero de texto llamado *temp.txt*.

En la primera línea del fichero está el dato (el valor medio de las temperaturas de los 31 días del mes) del mes de enero del primer año registrado, en la segunda línea está el dato del mes de febrero del mismo año, en la tercera el dato del mes de marzo del mismo año, etc. En la línea número 13 está escrito el dato del mes de enero del segundo año registrado, así sucesivamente. La última línea del fichero contiene el dato correspondiente al mes completo más reciente. Por ejemplo, si se lee el fichero a mediados del mes de junio de 2022, el último dato del fichero corresponde al mes de mayo de 2022.

Se desea calcular la media de los datos para cada uno de los 12 meses del año. Es decir, la media del mes de enero de todos los años registrados, la media del mes de febrero de todos los años registrados, etc.

Escriba un programa en C++ que abra el fichero de texto llamado *temp.txt* para lectura. Si no es posible abrir el fichero, el programa debe mostrar un mensaje en la consola indicándolo y terminar. El programa debe leer el fichero, calcular la media de los datos para cada uno de los 12 meses del año y mostrar en la consola los 12 valores medios calculados, en formato fijo con dos dígitos decimales.

Solución a la Pregunta 4

```cpp
#include <iostream>
#include <iomanip>
#include <fstream>
#include <vector>
#include <string>

std::string nombreFich = "temp.txt";
const std::string nombreMes[] = { "Enero", "Febrero", "Marzo", "Abril",
    "Mayo", "Junio", "Julio", "Agosto", "Septiembre", "Octubre",
    "Noviembre", "Diciembre" };

int main() {
  // Apertura del fichero para lectura
  std::ifstream inFich(nombreFich.c_str(), std::ios::in);
  if (!inFich) {
    std::cerr << "Error al abrir el fichero" << std::endl;
    return 0;
  }
  // Lectura del fichero y cálculo de la media mensual
  std::vector<double> acumuladoMensual(12, 0);
  std::vector<int> numDatosMensual(12, 0);
  unsigned int mes = 0;
  double dato;
  while (inFich >> dato) {
    acumuladoMensual[mes] += dato;
    numDatosMensual[mes]++;
    if (mes == 11)
      mes = 0;
    else
      mes++;
  }
  inFich.close();
  // Escritura del resultado en la consola
  std::cout << "Temperaturas medias mensuales\n";
  for (unsigned int i = 0; i < 12; i++)
    std::cout << nombreMes[i] << ":\t"
        << std::fixed << std::setprecision(2)
        << acumuladoMensual[i] / numDatosMensual[i] << "\n";
  return 0;
}
```

PREGUNTA 5 (3 puntos)

Consideremos un conjunto de círculos dibujados en el plano, definidos mediante las coordenadas cartesianas (X, Y) del centro y el radio de cada uno de los círculos. Además, cada círculo tiene un color, que puede ser uno de estos tres: rojo, azul o verde.

5.1 (1 punto) Escriba una función en C++ llamada `leeConsolaCirculo` que tiene el prototipo siguiente:

```
void leeConsolaCirculo(std::list<sCirc> &lCircs);
```

El argumento de la función es una referencia a una lista de elementos del tipo estructura `sCirc`, definido de la forma siguiente:

```
struct sCirc {
   double centroX, centroY, radio;
   std::string color;
};
```

La función debe escribir mensajes en la consola solicitando al usuario que éste introduzca los datos de un círculo (las coordenadas X e Y de su centro, su radio y su color) y debe leer esos datos.

Si el color introducido por el usuario es diferente de rojo, azul o verde, o bien el radio no es mayor que cero, la función finaliza su ejecución, devolviendo el control al programa principal.

En caso contrario, la función inserta al final de la lista pasada como referencia un nuevo elemento estructura con los datos del círculo especificado por el usuario.

5.2 (1 punto) Escriba una función en C++ llamada `getMaxRadio` que tiene el prototipo siguiente:

```
std::map<std::string, double> getMaxRadio(
                              std::list<sCirc> &lCircs );
```

La función debe examinar la lista pasada por referencia, a fin de encontrar el radio más grande de los círculos de cada color. La función devuelve un

mapa, cuya clave es un color y cuyo valor es el radio del círculo de mayor radio de ese color de entre los que se encuentran en la lista pasada por referencia a la función. Por tanto, la clave puede tomar los valores siguientes: rojo, azul y verde.

5.3 (1 punto) Escriba un programa en C++ que realice las acciones siguientes:

1. Declarar una constante global de tipo entero llamada `Ncirculos` y asignarle el valor 4.
2. Invocando la función `leeConsolaCirculo`, obtener del usuario los datos de `Ncirculos` círculos.
3. Escribir en la consola las coordenadas del centro, radio y color de cada uno de los `Ncirculos` introducidos por el usuario.
4. Invocando la función `getMaxRadio`, obtener el valor máximo del radio para cada uno de los colores de los círculos introducidos por el usuario y escribir dicho valor en la consola.
5. Terminar.

Solución a la Pregunta 5

```
#include <iostream>
#include <list>
#include <string>
#include <map>
#include <limits>

const int Ncirculos = 4;

struct sCirc {
  double centroX, centroY, radio;
  std::string color;
};


```

```
15
16 void leeConsolaCirculo(std::list<sCirc> &lCircs) {
17   sCirc circ;
18   const int leeCentro = 0, leeRadio = 1, leeColor = 2;
19   for (int lee = 0; lee <= leeColor; lee++) {
20     switch (lee) {
21     case leeCentro:
22       std::cout << "Introduzca las coordenadas X, Y del centro:\n";
23       std::cin >> circ.centroX >> circ.centroY;
24       break;
25     case leeRadio:
26       std::cout << "Introduzca el radio:\n";
27       std::cin >> circ.radio;
28       break;
29     case leeColor:
30       std::cout << "Introduzca el color (rojo, azul o verde):\n";
31       std::cin >> circ.color;
32       break;
33     }
34     if (std::cin.fail()) {
35       std::cin.clear();
36       std::cin.ignore(
37           std::numeric_limits<std::streamsize>::max(), '\n');
38       return;
39     }
40     if ((lee == leeRadio && circ.radio <= 0)
41         || (lee == leeColor && circ.color != "rojo"
42             && circ.color != "azul" && circ.color != "verde"))
43       return;
44   }
45   lCircs.push_back(circ);
46   return;
47 }
48
49
50 std::map<std::string, double> getMaxRadio(std::list<sCirc> &lCircs) {
51   std::map<std::string, double> mCirc;
52   std::list<sCirc>::iterator pL = lCircs.begin();
53   while (pL != lCircs.end()) {
54     std::map<std::string, double>::iterator pM;
55     pM = mCirc.find(pL->color);
56     if (pM == mCirc.end() || pM->second < pL->radio)
57       mCirc[pL->color] = pL->radio;
58     pL++;
59   }
60   return mCirc;
61 }
62
63
```

```
64
65 int main() {
66   std::list<sCirc> lCircs;
67   while (lCircs.size() < Ncirculos)
68     leeConsolaCirculo(lCircs);
69   std::list<sCirc>::iterator p = lCircs.begin();
70   while (p != lCircs.end()) {
71     std::cout << p->centroX << " " << p->centroY << " "
72         << p->radio << " "
73         << p->color << std::endl;
74     p++;
75   }
76   std::map<std::string, double> mCirc = getMaxRadio(lCircs);
77   // Escritura en consola
78   std::map<std::string, double>::iterator pM;
79   pM = mCirc.begin();
80   while (pM != mCirc.end()) {
81     std::cout << "Color " << pM->first
82         << ", Radio: " << pM->second << "\n";
83     pM++;
84   }
85   return 0;
86 }
```