LENGUAJES DE PROGRAMACIÓN

Solución al examen de Septiembre 2015

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 ALGOL 58 los identificadores pueden tener un longitud máxima de 8 caracteres.

B. (0.5 puntos) El lenguaje Java soporta un único tipo de número entero llamado **int**.

C. (0.5 puntos) El lenguaje Java no presenta el problema de las variables dinámicas perdidas.

D. (0.5 puntos) En C++, el resultado de la división entera 10/20 es 0.5.

E. (0.5 puntos) En C++ y Java, es obligatoria la existencia de un código de inicialización en la sentencia **for**.

F. (0.5 puntos) En un mapa, el tipo dominio y el tipo rango pueden ser diferentes.

Solución a la Pregunta 1

A	Falso	Véase la página 47 del texto base.
B	Falso	Véase la página 117 del texto base.
C	Verdadero	Véanse las páginas 126 y 127 del texto base.
D	Falso	Véase la página 192 del texto base.
E	Falso	Véase la página 274 del texto base.
F	Verdadero	Véase la página 435 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>

double y = 1.111;

int main () {
   double *x;
   x = &y;
   double v[3] = {2.222, 3.333, 4.666};
   std::cout << *x << std::endl;
   {
      double y = 8.898;
      std::cout << std::scientific
                << std::setprecision(2)
                << y
                << std::endl;
      std::cout << ::y
                << std::endl;
      std::cout << *x
                << std::endl;
      v[1] = y;
      x = &v[0];
   }
   x[0] = x[0] + 1;
   std::cout << *x   << std::endl;
   std::cout << v[0] << " "
             << v[1] << " "
             << v[2] << std::endl;
   x = x + 1;
   std::cout << x[0]   << std::endl;
   std::cout << *(x+1) << std::endl;
   return 0;
}
```

Solución a la Pregunta 2

```
1.111
8.90e+000
1.11e+000
1.11e+000
3.22e+000
3.22e+000 8.90e+000 4.67e+000
8.90e+000
4.67e+000
```

PREGUNTA 3 (1 punto)

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

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

int main(){
   std::vector<double> x;
   x.reserve(10);
   std::cout << std::boolalpha
             << x.empty()
             << std::endl;

   std::vector<double> y(4,5);
   x.assign(y.begin()+1, y.begin()+3);
   x.push_back(7);
   std::cout << x.capacity() << std::endl;

   for (unsigned int i = 0; i<x.size(); i++)
      std::cout << x[i] << " ";
   std::cout << std::endl;

   x.at(1) = 20;
   double a[] = {12, 22, 32};
   x.assign(a, a+3);
   for (unsigned i = 0; i<x.size(); i++)
      std::cout << x[i] << " ";
   std::cout << std::endl;

   x.pop_back();
   x.at(1) = 15;
   for (unsigned int i = 0; i<x.size(); i++)
      std::cout << x[i] << " ";

   return 0;
}
```

Solución a la Pregunta 3

```
true
10
5 5 7
12 22 32
12 15
```

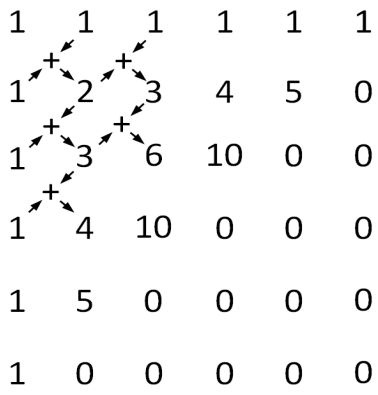
PREGUNTA 4 (2.5 puntos)

El triángulo de Pascal es una matriz de valores enteros que, ordenados por filas, representan los coeficientes binomiales. En la siguiente figura se muestra un triángulo de tamaño 6.

```
          1
        1   1
      1   2   1
    1   3   3   1
  1   4   6   4   1
1   5  10  10  5   1
```

Los laterales del triángulo tienen valor 1. El resto de los valores del triángulo se calculan como la suma de los dos valores que se encuentran inmediatamente encima del elemento a calcular.

Escriba un programa en C++ que calcule un triángulo de Pascal de un tamaño cualquiera N y lo muestre en la consola. Para ello, el programa debe usar un array bidimensional $N \times N$, en el cual ordenará los resultados de manera similar a como se muestra en la figura siguiente.



El programa debe realizar las acciones siguientes:

1. Declarar una constante global **int** llamada N y asignarle el valor 6.
2. Declarar un array bidimensional de **int**, de tamaño $N \times N$, llamado A.
3. Sea $a(i, j)$ el elemento situado en la fila i, columna j, de A. Realizar las asignaciones siguientes:

a) Asignar valor uno a los elementos de la columna cero:

$$a(i,0) = 1 \qquad \text{para } i = 0, \ldots, N-1$$

b) Asignar valor uno a los elementos de la fila cero:

$$a(0,j) = 1 \qquad \text{para } j = 0, \ldots, N-1$$

c) Asignar valor a los elementos que, no estando en la fila cero ni en la columna cero, están en la diagonal secundaria o situados sobre la diagonal secundaria:

$$a(i,j) = a(i-1,j) + a(i,j-1) \quad \text{para} \quad i > 0, j > 0 \quad \text{tales que} \quad i + j < N$$

4. Mostrar en la consola el triángulo de Pascal de tamaño N. Cada fila del triángulo debe escribirse en la consola en su correspondiente fila. Los elementos de una misma fila deben escribirse separados por un espacio en blanco. Por ejemplo, la salida correspondiente al triángulo de tamaño 6 deberá ser:

```
1
1 1
1 2 1
1 3 3 1
1 4 6 4 1
1 5 10 10 5 1
```

5. Terminar.

Solución a la Pregunta 4

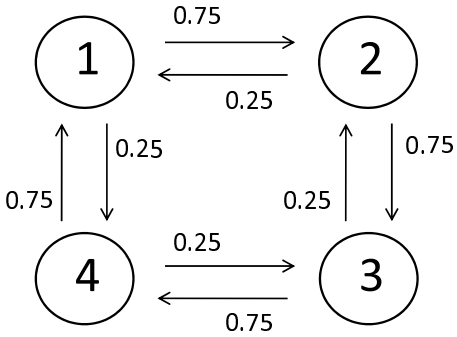
```
#include <iostream>

const int N = 6; // Tamaño del triángulo

int main() {
    int A[N][N];
    // Elementos columna 0
    for (int i=0; i<N; i++)
        A[i][0] = 1;
    // Elementos fila 0
    for (int j=0; j<N; j++)
        A[0][j] = 1;
    // Resto elementos triángulo de Pascal
    for (int i=1; i<N; i++)
        for (int j=1; j<N-i; j++)
            A[i][j] = A[i-1][j] + A[i][j-1];
    // Salida por consola triángulo
    for (int i=0; i<N; i++) {
        for (int j=0; j<=i; j++)
            std::cout << A[i-j][j] << " ";
        std::cout << std::endl;
    }
    return 0;
}
```

PREGUNTA 5 (2.5 puntos)

La figura muestra el diagrama de transición entre cuatro estados representados mediante los números 1, 2, 3 y 4. Las flechas representan las posibles transiciones entre estados. El número escrito junto a cada transición indica la probabilidad de que se produzca dicha transición.



Así, por ejemplo, si el sistema se encuentra en el estado 1, existe una probabilidad 0.75 de que pase al estado 2 y una probabilidad 0.25 de que pase al estado 4. Si se encuentra en el estado 2, pasará al estado 3 con probabilidad 0.75 y al estado 1 con probabilidad 0.25. Análogamente para las transiciones desde los estados 3 y 4.

Por otra parte, supongamos que el fichero de texto *random.txt* contiene una secuencia de números pseudoaleatorios.

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

1. Declarar una variable **int** llamada `estado`, la cual irá almacenando el valor del estado. Inicializar esta variable al valor uno.

2. Declarar una lista de **int** llamada `randomWalk` y añadir a la lista el valor de la variable `estado`.

3. Abrir el fichero de texto *random.txt* para lectura. Si se produce error, el programa deberá mostrar un mensaje en la consola indicándolo y terminar.

4. Ir leyendo uno a uno todos los números contenidos en el fichero. Para cada número leído, u_i, realizar las acciones siguientes:

   a) El valor de la variable `estado` indica el estado actual. Del estado actual salen dos posibles transiciones: una con probabilidad 0.25 y otra con probabilidad 0.75.

- Si $u_i \leq 0.25$, realizar la transición cuya probabilidad es 0.25.
- Si $u_i > 0.25$, realizar la transición cuya probabilidad es 0.75.

Asignar a `estado` el número que representa el estado destino de la transición, que pasa así a ser el estado actual.

b) Añadir el valor de `estado` al final de la lista `randomWalk`.

5. Cerrar el fichero de texto *random.txt*.
6. Mostrar en la consola todos los elementos de la lista `randomWalk`.
7. Aplicar a la lista `randomWalk` el algoritmo `count`, a fin de contar el número de veces que aparece en la lista el valor 4. Mostrar el resultado en la consola.
8. Terminar.

Solución a la Pregunta 5

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

const std::string nombreFichI = "random.txt";

int main() {
    // Apertura del fichero para lectura
    std::ifstream inFich(nombreFichI.c_str(), std::ios::in);
    if (!inFich) {
        std::cout << "ERROR al abrir fichero "
            << nombreFichI << std::endl;
        return 1;
    }
    // Lectura del fichero y transiciones
    int estado = 1;
    std::list<int> randomWalk;
    randomWalk.push_back(estado);
    while ( !inFich.eof() ) {
        double num;
        inFich >> num;
        if (inFich.eof()) break;
        switch ( estado ) {
        case 1:
            if (num <= 0.25) estado = 4; else estado = 2;
            break;
        case 2:
            if (num <= 0.25) estado = 1; else estado = 3;
            break;
        case 3:
            if (num <= 0.25) estado = 2; else estado = 4;
            break;
        case 4:
            if (num <= 0.25) estado = 3; else estado = 1;
            break;
        }
        randomWalk.push_back(estado);
    }
    // Cerrar el fichero
    inFich.close();
    // Mostrar en consola el contenido de la lista
    std::list<int>::iterator p = randomWalk.begin();
    while ( p != randomWalk.end() ) {
        std::cout << *p << " ";
        p++;
    }
    // Número elementos con valor 4
    std::cout << "\nNumero de elementos con valor 4: "
        << count(randomWalk.begin(), randomWalk.end(), 4)
        << std::endl;
    return 0;
}
```