

LENGUAJES DE PROGRAMACIÓN

Solución al examen de Junio 2015, Primera 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 una variable cuyo nombre sea NUM es de tipo entero.
- B. (0.5 puntos) En C++ y Java, el valor que se asigna a las constantes no puede ser el resultado de evaluar una expresión.
- C. (0.5 puntos) La expresión en notación prefija $*X + YZ$ es equivalente a la expresión $X + Y * Z$ en notación infija.
- D. (0.5 puntos) En C++ no se permite emplear expresiones numéricas como expresiones lógicas.
- E. (0.5 puntos) La excepción de C++ `std::bad_alloc` se puede producir al emplear el operador `new`.
- F. (0.5 puntos) El algoritmo de la STL

```
remove_copy(pBegin, pEnd, pResultBegin, val)
```

copia la secuencia origen en la secuencia destino, eliminando de la copia los elementos cuyo valor sea menor que `val`.

Solución a la Pregunta 1

- A** Verdadera Véase la página 43 del texto base.
- B** Falsa Véase la página 113 del texto base.
- C** Falsa Véase la página 171 del texto base.
- D** Falsa Véase la página 180 del texto base.
- E** Verdadera Véase la página 305 del texto base.
- F** Falsa Véase la página 528 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 x, y;

int main () {
    x = 1.06669;
    y = 4.62468;
    double *p;
    double a[2] = {1.233, 5.551};
    {
        double x = y;
        std::cout << std::scientific
                  << std::setprecision(3)
                  << x
                  << std::endl;
        std::cout << ::x << std::endl;
        p = &a[0];
    }
    p[0] = p[0] + 1.0;
    std::cout << a[0] << std::endl;
    p = p + 1;
    std::cout << p[0] << std::endl;
    std::cout << *(a+1) << std::endl;
    return 0;
}
```

Solución a la Pregunta 2

La salida por consola producida al ejecutar el programa es:

```
4.625e+000
1.067e+000
2.233e+000
5.551e+000
5.551e+000
```

PREGUNTA 3 (1 punto)

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

```
#include <iostream>
#include <stack>
#include <list>

int main(){
    std::stack <int> p;
    std::list <int> l;
    for (int i= 0; i<3; i++) {
        p.push(i);
        l.push_front(i);
    }
    std::list<int>::iterator it = l.begin();
    std::cout << p.size() << std::endl;
    std::cout << l.size() << std::endl;
    for (int i= 0; i<3; i++){
        std::cout << p.top() << std::endl;
        std::cout << *it << std::endl;
        it++;
        l.pop_front();
    }
    std::cout << p.size() << std::endl;
    std::cout << l.size() << std::endl;
}
```

Solución a la Pregunta 3

La salida por consola producida al ejecutar el programa es:

```
3
3
2
2
2
1
2
0
3
0
```

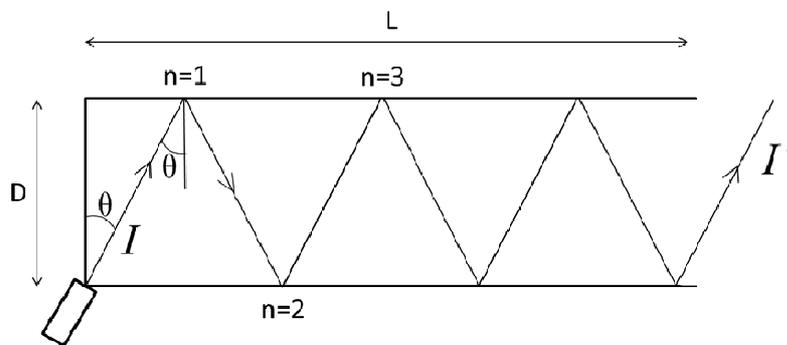
PREGUNTA 4 (2.5 puntos)

Un haz de luz de intensidad I , que es reflejado entre dos superficies reflectantes iguales, tiene después de n reflexiones una intensidad I' :

$$I' = I \cdot R^n \quad (1.1)$$

$$R = R_0 + (1 - R_0)(1 - \cos \theta)^5 \quad (1.2)$$

donde R_0 es la relación de reflexión y θ es el ángulo de incidencia del rayo, tal y como se muestra en la figura siguiente.



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

1. La intensidad del haz de entrada y la relación de reflexión deben ser constantes del programa con valor: $I = 100$, $R_0 = 0.7$.
2. Solicitar al usuario que éste introduzca por consola los valores de D y L . Comprobar que ambos son números mayores que cero y, en caso de que no lo sean, mostrar un mensaje de error y terminar.
3. Mostrar en la consola el número total de reflexiones y la intensidad de salida (I') para cada uno de los tres ángulos de incidencia siguientes $\theta = 60^\circ$, $\theta = 70^\circ$ y $\theta = 80^\circ$.

Obsérvese que el número de reflexiones va a depender de D , L y θ (véase nuevamente la figura). Una vez calculado dicho número, la intensidad de salida se calcula de las Ecs. (1.1) y (1.2).

4. Terminar.

Solución a la Pregunta 4

```

#include <iostream>
#include <cmath>

const double PI = 3.14159265359;

const double R0 = 0.7; // Relación de reflexión
const double I = 100; // Intensidad incidente

int main () {
    // Entrada de datos
    double D, L;
    std::cout << "Valor de D: ";
    std::cin >> D;
    std::cout << "Valor de L: ";
    std::cin >> L;
    if (D <= 0 || L <= 0 ) {
        std::cout << "Error: valor no valido"<< std::endl;
        return 0;
    }
    // Cálculo y salida de resultados
    for ( int theta = 60; theta <= 80; theta += 10) {
        // Ángulo expresado en radianes
        double theta_rad = theta*PI/180;
        // Avance del rayo entre dos reflexiones
        double u = D*tan(theta_rad);
        // Número de reflexiones
        int n = (int) (L/u);
        // Cálculo de R
        double R = R0 + (1-R0)*pow(1-cos(theta_rad),5);
        // Salida
        std::cout << "theta = " << theta << " grados\t" <<
            "n = " << n << "\t" << "I' = " << I*pow(R,n) << std::endl;
    }
    return 0;
}

```

PREGUNTA 5 (2.5 puntos)

Se desea estudiar cómo evoluciona en el tiempo el número N_t de bacterias presentes en un cultivo cuyo medio tiene una concentración de nutrientes C_t . El subíndice t representa el número de unidades de tiempo transcurridas desde que se inició la observación. Se va a emplear para ello el modelo siguiente:

- La velocidad de crecimiento per cápita de la colonia de bacterias en una unidad de tiempo, $(N_{t+1} - N_t)/N_t$, es proporcional a la concentración C_t de nutrientes en el medio en el instante t . Es decir:

$$\frac{N_{t+1} - N_t}{N_t} = m \cdot C_t \quad (1.3)$$

- Para producir un incremento de una unidad en la población, se consumen γ unidades de nutriente. Es decir:

$$\frac{C_{t+1} - C_t}{N_{t+1} - N_t} = -\gamma \quad (1.4)$$

Combinando las Ecs. (1.3) y (1.4), se obtiene el siguiente sistema de ecuaciones en diferencias:

$$N_{t+1} = (1 + m \cdot C_t) \cdot N_t \quad (1.5)$$

$$C_{t+1} = (1 - \gamma \cdot m \cdot N_t) \cdot C_t \quad (1.6)$$

donde $t = 0, 1, 2, 3, \dots$

Se propone realizar un programa en C++ que evalúe las Ecs. (1.5) y (1.6), para $t = 0, 1, \dots, T_{\text{fin}}$, siendo conocidos: T_{fin} , los parámetros m y γ , y las condiciones iniciales N_0 y C_0 .

Escriba el código C++ indicado a continuación.

5.a (0.5 puntos) Escriba una función que evalúe las Ecs. (1.5) y (1.6), y cuyo prototipo sea:

```
void logis( double m, double gamma,
           double &N, double &C );
```

Los parámetros m y γ son pasados por valor, mientras que N y C son pasados por referencia: los valores N_t y C_t son reemplazados en el cuerpo de la función por N_{t+1} y C_{t+1} , respectivamente.

5.b (1 punto) Escriba una función cuyo prototipo sea:

```
void escribeFich( std::string nombreFich,
                 std::vector<double> &vN )
    throw (std::invalid_argument);
```

y que realice las acciones siguientes:

1. Abrir para escritura el fichero de texto cuyo nombre está almacenado en la variable `nombreFich`. Si no es posible abrir el fichero, la función debe lanzar una excepción.
2. Escribir en el fichero una columna de números: el contenido del vector `vN`. Los números deben escribirse en formato fijo, con 4 dígitos decimales.
3. Cerrar el fichero.
4. Devolver el control.

5.c (1 punto) Escriba el programa principal, que debe realizar las acciones siguientes:

1. Declarar las constantes siguientes:

`Tfin=100`, de tipo **int**.

`m=0.1`, `gamma=0.02`, `N0=1.0` y `C0=1.0`, de tipo **double**.

2. Declarar un vector de tipo **double** llamado `vN` y almacenar en él los valores $N_0, N_1, \dots, N_{T_{\text{fin}}}$ obtenidos de evaluar las Ecs. (1.5) y (1.6). Los valores de N_{t+1} y C_{t+1} deben obtenerse invocando la función definida en el Apartado 5.a.
3. Escribir en la consola los valores t, N_t y C_t , para $t = 0, 1, \dots, T_{\text{fin}}$.
4. Escribir el contenido del vector en un fichero de texto llamado *poblacion.txt*, invocando para ello la función definida en el Apartado 5.b. Si la función lanza una excepción, debe capturarse la excepción mostrándose un mensaje de aviso en la consola en el que se indique que no ha sido posible abrir el fichero.
5. Terminar.

Solución a la Pregunta 5

```

#include <iostream>
#include <iomanip>
#include <fstream>
#include <string>
#include <stdexcept>
#include <vector>

const std::string nombreFich = "poblacion.txt";
const int Tfin = 100;
const double m = 0.1;
const double gamma = 0.02;
const double N0 = 1.0;
const double C0 = 1.0;

void logis(double m, double gamma, double &N, double &C) {
    double Nt1 = (1 + m * C) * N;
    C = (1 - gamma * m * N) * C;
    N = Nt1;
    return;
}

void escribeFich(std::string nombreFich,
                std::vector<double> &vN )
                throw (std::invalid_argument) {
    // Apertura del fichero para escritura
    std::ofstream outFich(nombreFich.c_str(),
                          std::ios::out | std::ios::trunc);
    if (!outFich)
        throw std::invalid_argument("Error abriendo fichero");
    // Escritura de datos
    for (int i=0; i<vN.size(); i++)
        outFich << std::fixed << std::setprecision(4) <<
                vN[i] << std::endl;
    // Cierra fichero
    outFich.close();
    return;
}

```

```
int main() {
    std::vector<double> vN;
    // Evaluar ecuaciones
    vN.push_back(N0);
    double C = C0;
    std::cout << 0 << "\t" << vN[0] << "\t" << C << std::endl;
    for (int i=0; i<Tfin; i++) {
        vN.push_back(vN[i]);
        logis(m, gamma, vN[i+1], C);
        std::cout << i+1 << "\t" << vN[i+1] << "\t" << C << std::endl;
    }
    // Escritura en fichero
    try {
        escribeFich(nombreFich, vN);
    } catch (std::invalid_argument exc) {
        std::cout << exc.what() << std::endl;
    }
    return 0;
}
```