

Prob.1. Considera la siguiente función que evalúa una expresión guardada en un árbol y que contiene en las hojas símbolos que pueden ser evaluados (se obtiene su valor) a través de la función `ValorSimbolo`.

```
float EvaluarExpresion (Arbol & arb)
{
    float aux1, aux2, total, aux;
    char c_aux;

    total = 0;

    if (!arb.ArbolVacio())
    {
        arb.Informacion (c_aux);

        //si el nodo es un operando
        if (!EsOperador (c_aux) )
        {
            total = ValorSimbolo (c_aux);
        }
        //si el nodo es un operador
        else
        {
            // Se calcula el resultado de la evaluacion de los subarboles hijos
            aux1 = EvaluarExpresion (arb.HijoIzdo (), t);
            aux2 = EvaluarExpresion (arb.HijoDcho (), t);
            total = Operar (aux1, aux2, c_aux);
        }
    }
    return total;
}
```

Los prototipos de las funciones utilizadas en la función son:

```
float ValorSimbolo (char simbolo);
    Obtiene el valor real de un símbolo

float Operar (float x, float y, char op);
    Opera 'x' e 'y' con la operación 'op'
```

Considera también la función que dado un `float` lo transforma en un `string`

```
string convertir_a_cadena(float num);
```

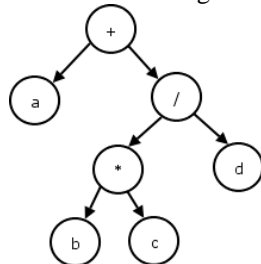
Reescribe la función `EvaluarExpresion` para que devuelva a través de un parámetro, una cadena que contenga la expresión instanciada con los valores de evaluación.

Nota: para que quede claro el orden de ejecución deberás usar paréntesis.

Ejemplo:

Cadena introducida con la cual se ha generado el árbol: $a+(b*c)/d$

Árbol resultante:



Valores de las variables: $a= 3.4$, $b=2$, $c=2.1$, $d=3$

Cadena devuelta en el string de `EvaluarExpresion`: $(3.4+((2*2.1)/3))$

Prob.1.

```
float EvaluarExpresion (Arbol & arb, string & cad)
{
    float aux1, aux2, total, aux;
    char c_aux;
    string cad1, cad2;

    total = 0;

    if (!arb.ArbolVacio())
    {
        arb.Informacion (c_aux);

        //si el nodo es un operando
        if (!EsOperador (c_aux) )
        {
            total = ValorSimbolo (c_aux);
            cad = convertir_a_cadena (total);
        }
        //si el nodo es un operador
        else
        {
            // Se calcula el resultado de la evaluacion de los subarboles hijos
            aux1 = EvaluarExpresion (arb.HijoIzdo (), cad1);
            aux2 = EvaluarExpresion (arb.HijoDcho (), cad2);
            total = Operar (aux1, aux2, c_aux);
            cad = "(" + cad1 + c_aux + cad2 + ")" ;
        }
    }
    return total;
}
```

Prob.2. Barajar una baraja usando el método del crupier consiste en dividirla en dos mazos con un nº de cartas similar e ir intercalando una carta de un mazo y otra del otro, repitiendo este proceso varias veces. Usando las clases Carta y Pila cuya interfaz teneis a continuación, y las estructuras auxiliares que creas conveniente, implementa la función BarajarCrupier usando el método del crupier.

Clase CARTA:

```
class Carta
{
public:
    Carta ();
    bool ponerCarta (char, int);
    //devuelve el palo de la carta
    char paloCarta ();
    //devuelve el numero de la carta
    int numeroCarta ();
    //muestra por pantalla la carta
    void MostrarCarta ();
private:
    ...
};
```

Clase PILA:

```
class Pila
{
public:
    Pila ();
    bool Apilar (Carta);
    bool Desapilar ();
    bool CimaPila (Carta &);
    bool PilaVacía ();
    void MostrarPila ();
    int NElementosPila ();
private:
    ...
};
```

Para realizar el algoritmo supondremos la existencia de la función 'int aleatorio (int max);' que determina números aleatorios entre '0' y 'max - 1'.

Prob.2.

```
void Barajar (Pila & baraja)
{
    int num_veces, num, i;    Pila p_aux1, p_aux2;    Carta x, y;

    num_veces = 5 + aleatorio (10);
    for (i = 0; i < num_veces; i++)
    {
        // La mitad de la baraja sera 24 +/- 5 cartas
        num = 19 + aleatorio (11);

        // Separo las cartas en dos montones que guardo en p_aux1 y p_aux2
        for (j = 0; j < num; j++)
        {
            baraja.CimaPila (x);
            p_aux1.Apilar (x);
            baraja.Desapilar ();
        }
        j = 0;
        while (baraja.CimaPila (x) )
        {
            p_aux2.Apilar (x);
            baraja.Desapilar ();
        }

        // Cojemos una carta de cada monton y la ponemos en la baraja
        while (p_aux1.CimaPila (x) && p_aux2.CimaPila (y) )
        {
            baraja.Apilar (x);
            baraja.Apilar (y)
            p_aux1.Desapilar ();
            p_aux2.Desapilar ();
        }

        // Ponemos las cartas que nos queden en un mazo u otro en el mazo principal
        while (p_aux1.CimaPila (x) )
        {
            baraja.Apilar (x);
            p_aux1.Desapilar ();
        }
        while (p_aux2.CimaPila (x) )
        {
            baraja.Apilar (x);
            p_aux2.Desapilar ();
        }
    }
    return;
```

}

Prob.3. Supongamos una cierta clase 'Biblioteca' cuya interfaz es:

```
class Biblioteca
{
    public:
        bool LeerBiblioteca (string);
        void MostrarBiblioteca ();
        void OrdenarBiblioteca ();
        void BuscarPorAutor (string);
        void BuscarPorTitulo (string);
        int Numero ();

    private:
        // contenedor de libros
        typedef Libro Vector[MAX];
        Vector info;
        int num;
        void OrdenarSeleccion (int &, int &);
        int BusquedaDicotomica (string, int &);
        int BusquedaSecuencial (string, int &);
};
```

Basada, como se puede apreciar, en la clase 'Libro':

```
class Libro
{
    public:
        void CambiarAutor (string a);
        void CambiarTitulo (string t);
        void CambiarReferencia (int r);
        void CambiarPrecio (int p);
        string Autor ();
        string Titulo ();
        void MostrarLibro ();

    private:
        string titulo;
        string autor;
        int ref;
        float precio;
};
```

Realizar el método `LeerBiblioteca` al que se le pasa el nombre de un fichero que contiene información de libros y rellena la biblioteca con los libros contenidos en el fichero.

El formato del fichero es el siguiente:

```
Nombre del libro
Nombre y apellidos del autor
Referencia Precio
```

Por ejemplo:

```
Conan
Howard, Robert E.
1 8.35
Las Enseñanzas de Don Juan
Castaneda, Carlos
2 10.5
```

Prob.2.

```
bool Biblioteca::LeerBiblioteca (string nom)
{
    ifstream f;
    string cad;
    int ref;
    float prec;
    bool exito;

    num = 0;
    f.open (nom.c_str () );
    if (!f)
        exito = false;
    else
    {
        exito = true;

        while (getline (f, cad) )
        {
            info[num].CambiarTitulo (cad);
            getline (f, cad);
            info[num].CambiarAutor (cad);
            f >> ref;
            info[num].CambiarReferencia (ref);
            f >> prec;
            info[num].CambiarPrecio (prec);
            f.ignore ();
            num++
        }
    }
}
```

```
    }  
  }  
  return;  
}
```