

# Estructuras de Datos

**ELABORADO POR**

**JORGE EDUARDO CARRIÓN VIRAMONTES**

Instituto Tecnológico de Morelia  
Departamento de Sistemas y Computación  
Av. Tecnológico # 1500  
Col. Lomas de Santiaguito  
Teléfono (443) 312 15 70

**ENERO, 2016**

## Operadores y Funciones

1. Descargue el paquete **Dev-Cpp 5.11 TDM-GCC 4.9.2 Setup.exe**, disponible en las descargas de la página.
2. Ejecute el programa para su instalación.
3. Localice el ejecutable **devcpp.exe** y ejecútelo.
4. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
5. En el nuevo documento transcriba el código del siguiente programa, en el cual se realiza la operación de desplazamiento a la derecha de los bits de un número entero, que equivale a una división entre dos con redondeo hacia abajo:

```
#include<iostream>
#include<cmath>
using namespace std;
int main(void) {
    int valor;
    cout<<"Division entre 2 mediante el desplazamiento de bits a la derecha ."<<endl<<endl;
    while (true) {
        cout<<"Dame un numero: ";
        cin>>valor;
        if (valor==0) break;
        while (abs(valor)!=1) {
            cout<<"El valor "<<valor<<" se va a desplazar una posicion a la derecha.";
            valor=valor>>1;
            cout<<endl<<"Convirtiendose ahora en el valor: ";
            cout<<valor<<endl<<endl;
        }
    }
}
```

6. Ahora localice el icono de **Compilar** que está en la barra de herramientas y oprímalo, asígnele el nombre de **ejercicio 1A** y elija el destino que guste para guardarlo.
7. Si su programa no contiene errores aparecerá el siguiente mensaje en la parte inferior que inicia con: **Compilation results...**
8. Ahora localice el siguiente icono en la barra de herramientas que corresponde a **Ejecutar** y oprímalo.
9. En la ventana emergente deberá aparecer el texto que solicita **Dame un numero:**
10. Si escribe **8**, obtendrá sucesivamente los valores **4**, **2** y **1**.
11. Y si escribe un número negativo, por ejemplo **-65**.
12. Aparecerá una serie de líneas que nos indican los números que se obtienen de ir corriendo un bit a la derecha: **-33**, **-17**, **-9**, **-5**, **-3**, **-2** y termina en **-1**.

13. Pruebe con otros valores enteros.
14. Si proporciona un número que no sea entero o una cadena de caracteres el programa hace la conversión automática a enteros.
15. Si queremos hacer **tronar** al programa podemos poner una cantidad menor que el mínimo aceptable que es **-2,147,483,648**; o mayor que el máximo posible que es **2,147,483,647**.
16. Escriba **0** para terminar el programa y oprima **intro** para cerrar la ventana de ejecución.
17. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
18. A continuación transcriba el siguiente ejemplo en el que se definen dos funciones sencillas con argumentos enteros y un programa principal que las utiliza:

```
#include<iostream>
#include <cmath>
using namespace std;

int maximo (int val1, int val2) {
    if (val1 > val2) return val1;
    else return val2;
}

double raiz (int num) {
    return sqrt(num);
}

int main(void) {
    int A,B,C;
    cout<<endl<<endl<<"Ejemplo de Funciones"<<endl<<endl;
    cout<<"proporciona un número entero: ";
    cin>>A;
    cout<<"proporciona otro entero: ";
    cin>>B;
    cout<<"el maximo de "<<A<<" y "<<B<<" es: "<<maximo(A,B)<<endl;
    cout<<"proporciona un entero para la raiz: ";
    cin>>C;
    cout<<"la raiz cuadrada de "<<C<<" es: "<<raiz(C)<<endl;
}
```

19. Ahora localice el icono de **Compilar** que está en la barra de herramientas y oprímalo, asígnele el nombre de **ejercicio 1B** y elija el destino que guste para guardarlo.
20. Ahora localice el siguiente icono en la barra de herramientas que corresponde a **Ejecutar** y oprímalo.
21. Pruebe con diferentes valores numéricos, incluso si proporciona números con decimales la aplicación ignora los decimales y conserva la parte entera para ejecutar las funciones.
22. Ahora vamos a crear una nueva versión de este programa para permitirle utilizar otros tipos de datos sin tener que crear otras versiones de funciones:

23. En el menú **Archivo** seleccione la opción **Nuevo** y a continuación **Archivo Fuente**.
24. Copie completamente el código del ejemplo anterior en esta nueva ventana y edítelo para que coincida con el siguiente código (los cambios aparecen resaltados):

```
#include<iostream>
#include <cmath>
#include <string>
using namespace std;

template < class t>
    t maximo (t val1, t val2) {
        if (val1 > val2) return val1;
        else return val2;
    }

template < class t>
double raiz (t num) {
    return sqrt(num);
}

int main(void) {
    int A,B,C;
    string F,G;
    float H;
    cout<<endl<<endl<<"Ejemplo de Plantillas de Funciones"<<endl;
    cout<<"proporciona un numero entero: ";
    cin>>A;
    cout<<"proporciona otro entero: ";
    cin>>B;
    cout<<" el maximo de "<<A<<" y "<<B<<" es: "<<maximo(A,B)<<endl;
    cout<<"proporciona una cadena: ";
    cin>>F;
    cout<<"proporciona otra cadena: ";
    cin>>G;
    cout<<" el maximo de "<<F<<" y "<<G<<" es: "<<maximo(F,G)<<endl;
    cout<<endl<<"proporciona un entero para la raiz: ";
    cin>>C;
    cout<<"la raiz cuadrada de "<<C<<" es: " <<raiz(C)<<endl;
    cout<<endl<<"proporciona un flotante para la raiz: ";
    cin>>H;
    cout<<"la raiz cuadrada de "<<H<<" es: " <<raiz(H)<<endl;
}
```

25. Ahora localice el icono de **Compilar** que está en la barra de herramientas y oprímalo, asígnele el nombre de **ejercicio 1C** y elija el destino que guste para guardarlo.
26. Ahora localice el siguiente icono en la barra de herramientas que corresponde a **Ejecutar** y oprímalo.

27. Proporcione los datos que se solicitan y observe que las mismas funciones, definidas como plantillas pueden trabajar correctamente con distintos tipos de datos.
28. Fin de la Práctica.