

## Tarea # 1

### Puntero Simple:

Un puntero es un objeto el cual hace referencia a otro valor almacenado en la memoria de una computadora. Estos pueden ser declarados como cualquier tipo de variable normal, pueden ser de tipo entero, cadena, carácter, booleano, etc; y pueden ser diferenciados de los otros por un (\*) como prefijo o sufijo. Por ejemplo: "int a\*;" Estos hacen referencia a un espacio de memoria el cual, al querer consultarlo utilizando el operador ampersand (&) antepuesto a la variable, aparecerá un numero en notación hexadecimal y este puede ser un numero cualquiera en el cual se haya guardado la información.

### Puntero Doble:

Un puntero doble es un objeto el cual hace referencia a otro objeto que hace referencia a un valor almacenado en memoria. La asignación hecha en un puntero se llama asignación indirecta, y esta altera el valor de la variable a la cual se le apunta. Para modificar el valor de una variable por medio de un puntero se hace de la siguiente manera. Asumiendo que ya tenemos una variable declarada y un valor asignado: "\*\*punterodoble = 5". De esta forma solo se altera el valor original al cual la variable apunta. Esto no afecta al valor que almacena la variable, la cual es la dirección de memoria del puntero simple.

### Macros:

Una macroinstrucción es una serie de instrucciones que se almacenan para que se puedan ejecutar de manera secuencial mediante una sola llamada. Tanto variables como instrucciones pueden abreviarse usando macros. Estas pueden ser invocadas en todo el código por una palabra clave, la cual se define al principio del mismo. Para ello se utiliza la palabra reservada define seguido de la instrucción que quiera definir. Una sentencia for puede ser abreviada de la siguiente forma:

"#define FOR(A<B) for(A=0; A<B;A++)", mandando a llamarlo en el código de la forma "int i; FOR(i,5){ }" la cual seria equivalente a "int i; for(i=0;i<5;i++){ }".

### Ejemplo Práctico:

Se utilizó una macroinstrucción y se definió de la manera:

```
#include <iostream>
#define ORIGINAL 20

using namespace std;

int main()
```

Se declaró un a variable normal, una de puntero simple, y otra de puntero doble; todas de tipo entero

```
int original;
int *puntero;
int **punterodoble;
```

Se le asigno valores a cada variable, original paso a ser el valor 20 definido en ORIGINAL, \*puntero apunta a la dirección de memoria de original y \*\*punterodoble a la dirección de memoria del apuntador simple.

```
original = ORIGINAL;

cout<<"El valor de original: "<<original<<endl;

puntero = &original; /* puntero apunta a original */
punterodoble = &puntero;

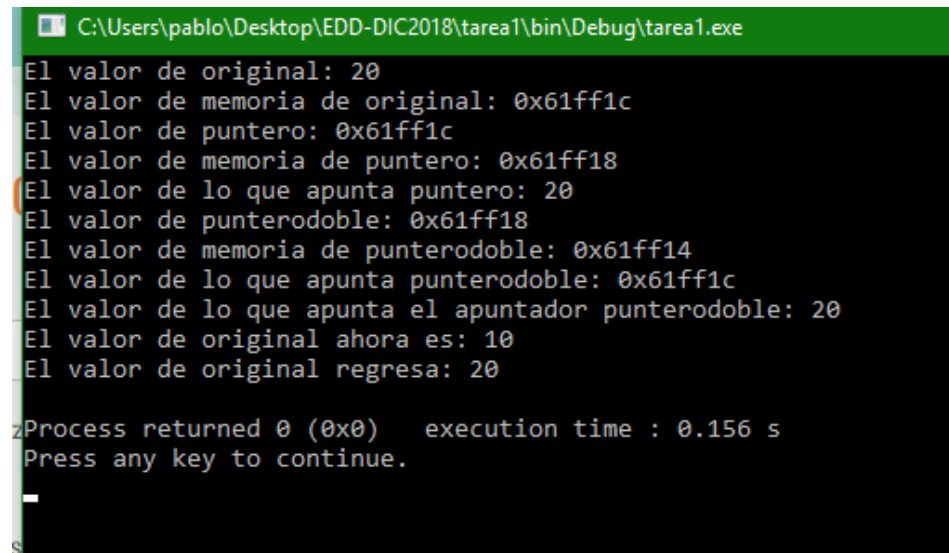
cout<<"El valor de memoria de original: "<<&original<<endl;
cout<<"El valor de puntero: "<<puntero<<endl;
cout<<"El valor de memoria de puntero: "<<&puntero<<endl;
cout<<"El valor de lo que apunta puntero: "<<*puntero<<endl;
cout<<"El valor de punterodoble: "<<punterodoble<<endl;
cout<<"El valor de memoria de punterodoble: "<<&punterodoble<<endl;
cout<<"El valor de lo que apunta punterodoble: "<<*punterodoble<<endl;
cout<<"El valor de lo que apunta el apuntador punterodoble: "<<**punterodoble<<endl;

*puntero = 10; /* el valor a lo que apunta puntero, lo cambiamos de 20 a 10

cout<<"El valor de original ahora es: "<<original<<endl;

**punterodoble = 20;

cout<<"El valor de original regresa: "<<original<<endl;
```



```
C:\Users\pablo\Desktop\EDD-DIC2018\tarea1\bin\Debug\tarea1.exe
El valor de original: 20
El valor de memoria de original: 0x61ff1c
El valor de puntero: 0x61ff1c
El valor de memoria de puntero: 0x61ff18
El valor de lo que apunta puntero: 20
El valor de punterodoble: 0x61ff18
El valor de memoria de punterodoble: 0x61ff14
El valor de lo que apunta punterodoble: 0x61ff1c
El valor de lo que apunta el apuntador punterodoble: 20
El valor de original ahora es: 10
El valor de original regresa: 20

Process returned 0 (0x0)   execution time : 0.156 s
Press any key to continue.
-
```