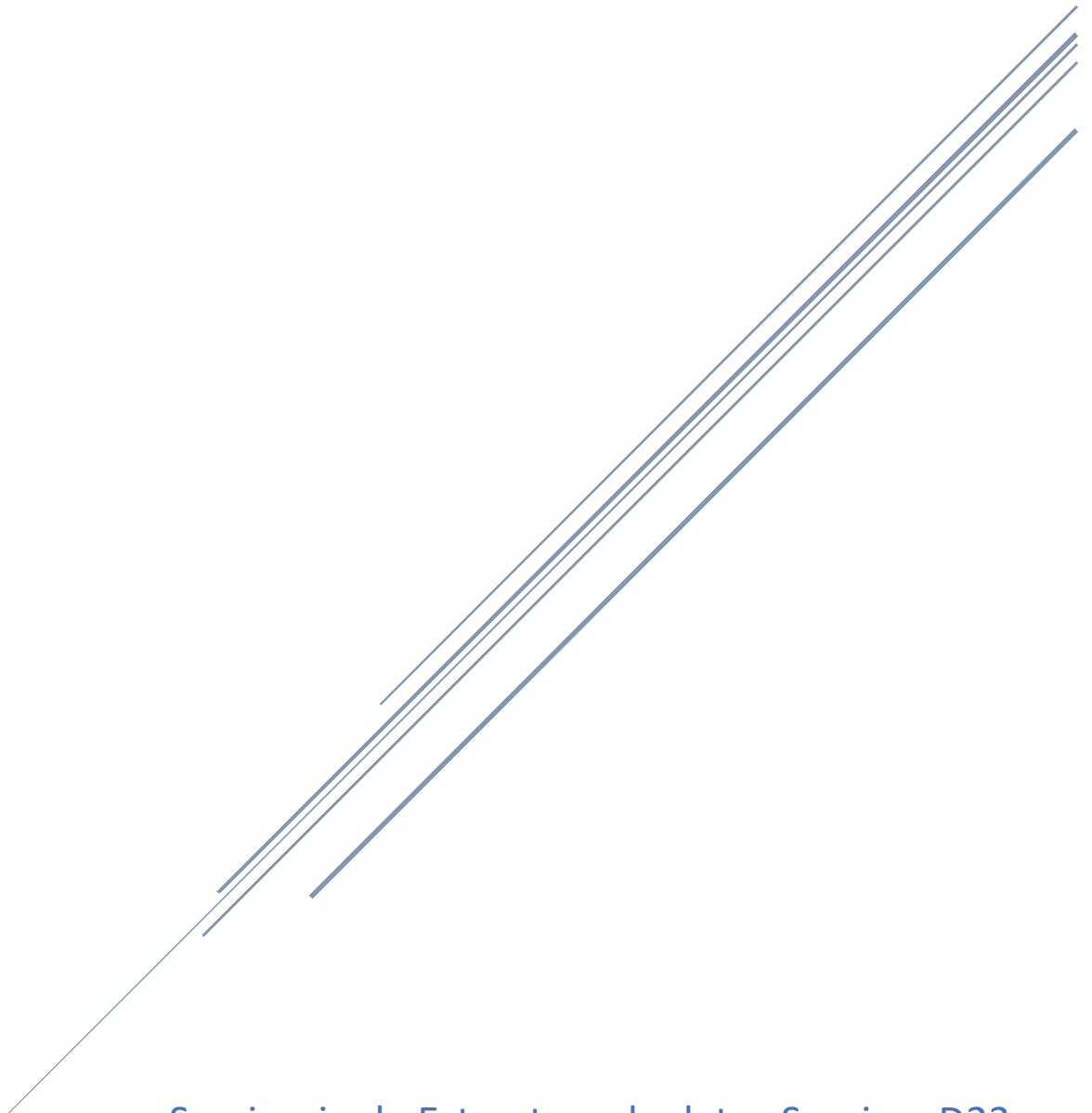


# PRACTICA 1

## Punteros



Seminario de Estructura de datos Seccion D23  
Rafael Villagomez Vega

#### Objetivos:

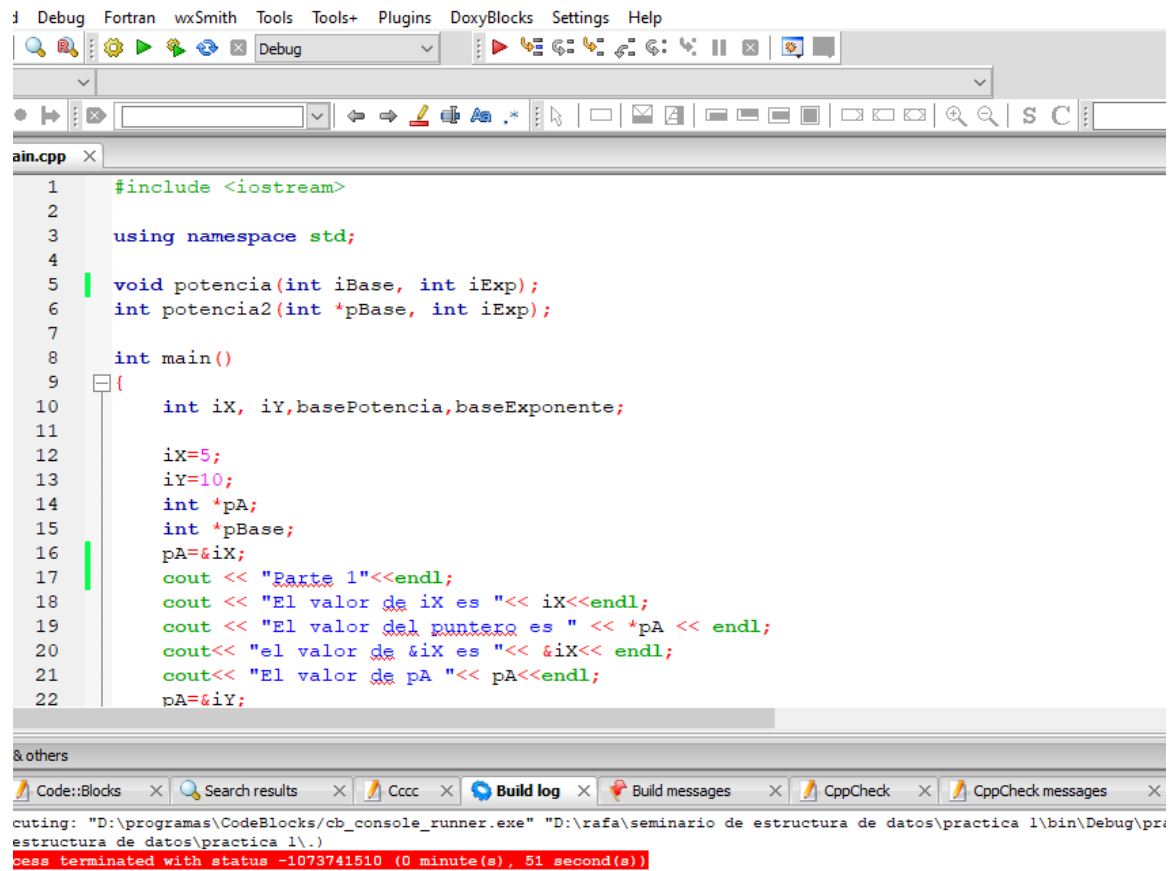
- Familiarizarse con el concepto de puntero.
- Familiarizarse con la declaración de punteros.
- Entender el uso del operador de dirección (&).
- Entender el uso del operador de indirección (\*).
- Comprender que un apuntador puede ser usado para trabajar con diferentes variables a través del tiempo, una a la vez.
- Utilizar funciones con argumentos de salida mediante el uso de punteros.

#### Descripción:

El objetivo del programa es el aprendizaje del concepto de punteros (o apuntadores), trabajar con punteros al asignarles dirección a la memoria donde haya valores asignados a una variable y además, trabajar en conjunto con las funciones.

#### Explicación:

En esta practica se aprendió a definir correctamente un puntero y como darle dirección. Los puntos 6 y 7 imprimen lo mismo porque están guardados en la misma sección de la memoria que tiene de valor 5, 8 y 9 imprimen lo mismo porque también están guardados en la misma sección de la memoria, aunque estos no tiene un valor asignado específica, solo imprimen basura. Lo mismo aplica en los puntos subsecuentes



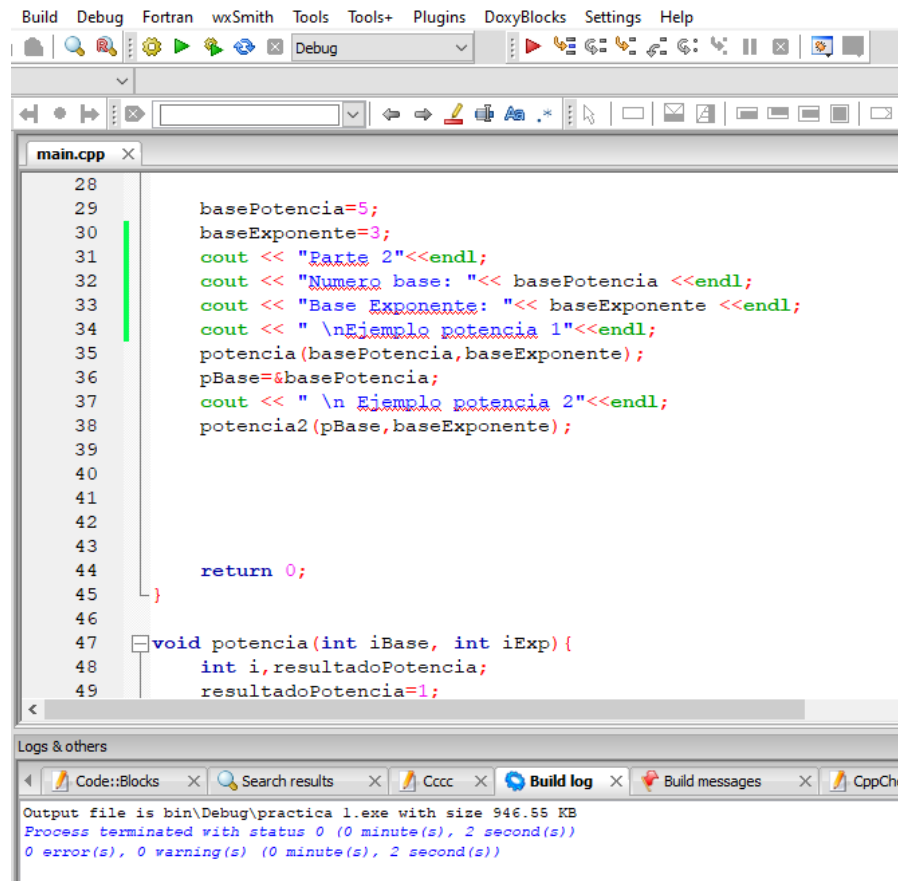
The screenshot shows the Code::Blocks IDE interface. The main editor window displays a C++ file named `ain.cpp` with the following code:

```
1  #include <iostream>
2
3  using namespace std;
4
5  void potencia(int iBase, int iExp);
6  int potencia2(int *pBase, int iExp);
7
8  int main()
9  {
10     int iX, iY, basePotencia, baseExponente;
11
12     iX=5;
13     iY=10;
14     int *pA;
15     int *pBase;
16     pA=&iX;
17     cout << "Parte 1"<<endl;
18     cout << "El valor de iX es "<< iX<<endl;
19     cout << "El valor del puntero es " << *pA << endl;
20     cout<< "el valor de &iX es "<< &iX<< endl;
21     cout<< "El valor de pA "<< pA<<endl;
22     pA=&iY;
```

Below the editor, the 'Run' window is open, showing the output of the program. The output is as follows:

```
cuting: "D:\programas\CodeBlocks\cb_console_runner.exe" "D:\rafa\seminario de estructura de datos\practica 1\bin\Debug\pr
estructura de datos\practica 1\."
Press terminated with status -1073741510 (0 minute(s), 51 second(s))
```

ks 13.12

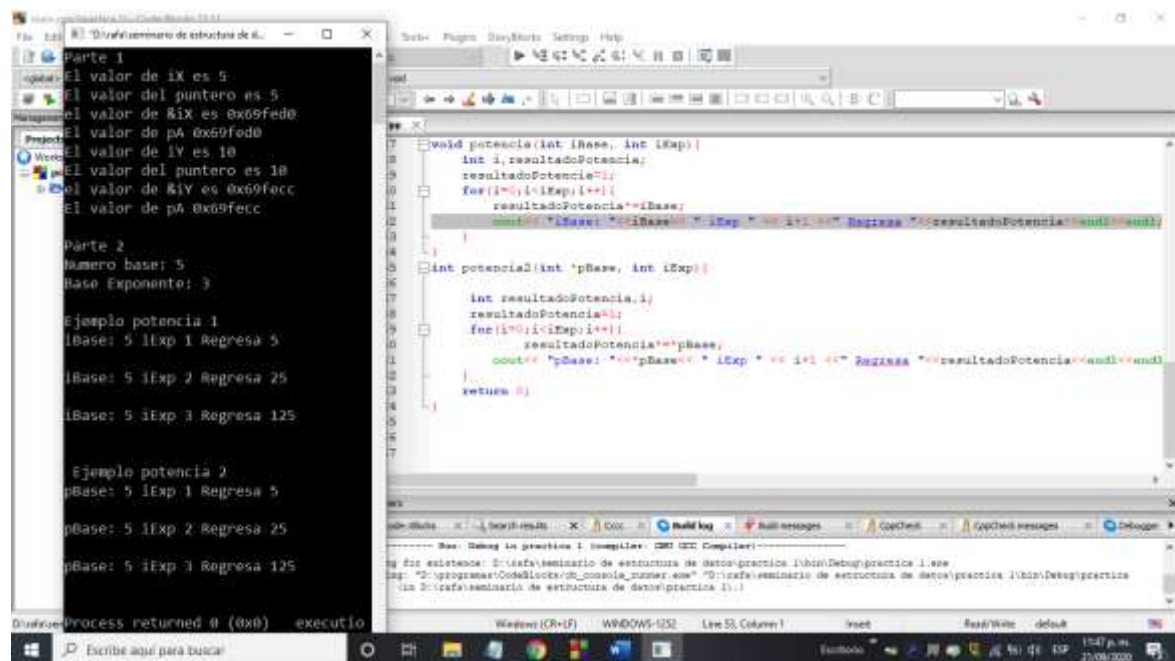


The screenshot shows a C++ IDE with the following components:

- Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help** menu bar.
- main.cpp** editor showing the following code:

```
28
29     basePotencia=5;
30     baseExponente=3;
31     cout << "Parte 2"<<endl;
32     cout << "Numero base: " << basePotencia <<endl;
33     cout << "Base Exponente: " << baseExponente <<endl;
34     cout << " \nEjemplo potencia 1"<<endl;
35     potencia(basePotencia,baseExponente);
36     pBase=&basePotencia;
37     cout << " \n Ejemplo potencia 2"<<endl;
38     potencia2(pBase,baseExponente);
39
40
41
42
43
44     return 0;
45 }
46
47 void potencia(int iBase, int iExp){
48     int i,resultadoPotencia;
49     resultadoPotencia=1;
```
- Logs & others** panel showing the build log:

```
Output file is bin\Debug\practica 1.exe with size 946.55 KB
Process terminated with status 0 (0 minute(s), 2 second(s))
0 error(s), 0 warning(s) (0 minute(s), 2 second(s))
```



The screenshot shows a C++ IDE with the following components:

- Partes** menu bar.
- potencia.cpp** editor showing the following code:

```
17 void potencia(int iBase, int iExp){
18     int i,resultadoPotencia;
19     resultadoPotencia=1;
20     for(i=0;i<iExp;i++){
21         resultadoPotencia*=iBase;
22         cout<<"*"<<iBase<<"*"<<iExp<<" = "<<i+1<<" Regresa "<<resultadoPotencia<<endl<<endl;
23     }
24 }
25
26 int potencia2(int *pBase, int iExp){
27
28     int resultadoPotencia,i;
29     resultadoPotencia=1;
30     for(i=0;i<iExp;i++){
31         resultadoPotencia**pBase;
32         cout<<"*"<<pBase<<"* "<<i+1<<" Regresa "<<resultadoPotencia<<endl<<endl;
33     }
34     return 0;
35 }
```
- Output** panel showing the program's output:

```
Parte 1
El valor de iX es 5
El valor del puntero es 5
El valor de &iX es 0x69fed0
El valor de pA 0x69fed0
El valor de iY es 10
El valor del puntero es 10
El valor de &iY es 0x69fec0
El valor de pA 0x69fec0

Parte 2
Numero base: 5
Base Exponente: 3

Ejemplo potencia 1
iBase: 5 iExp 1 Regresa 5
iBase: 5 iExp 2 Regresa 25
iBase: 5 iExp 3 Regresa 125

Ejemplo potencia 2
pBase: 5 iExp 1 Regresa 5
pBase: 5 iExp 2 Regresa 25
pBase: 5 iExp 3 Regresa 125
```
- Debug Console** panel showing the process return code: `Process returned 0 (0x0) execution`