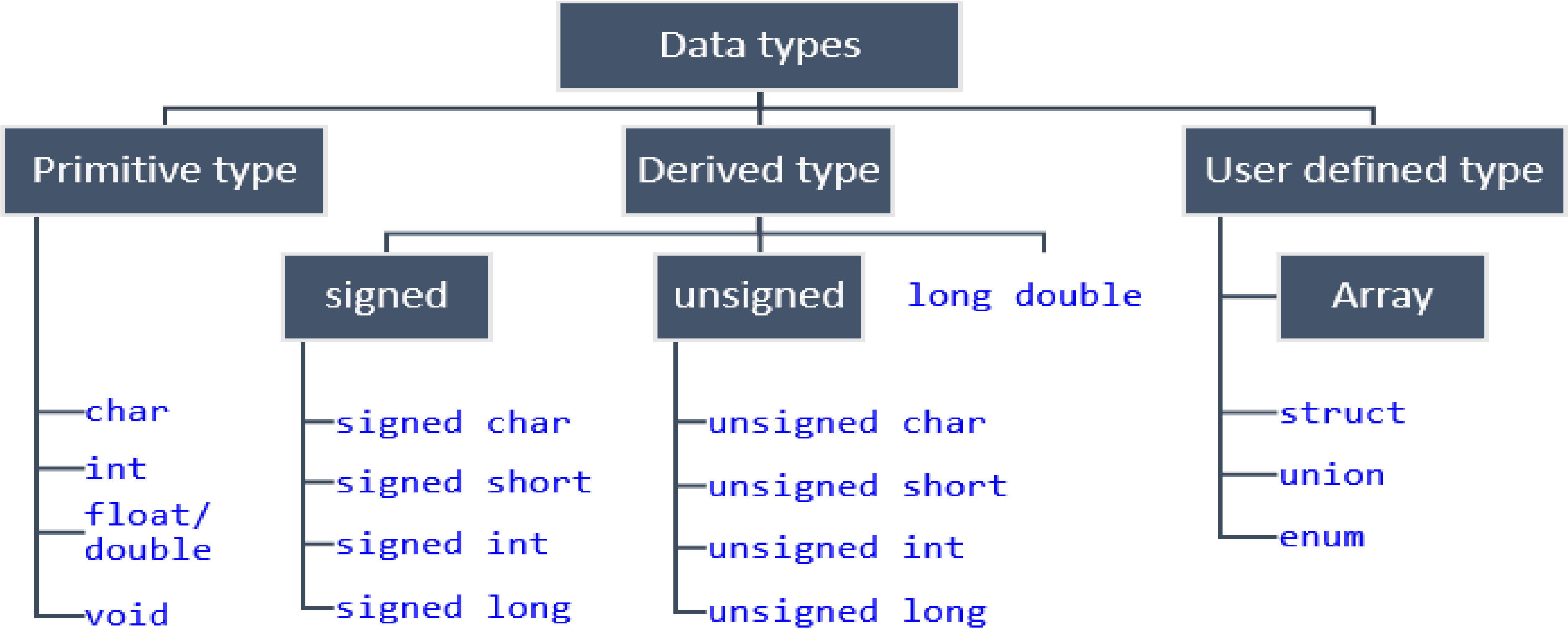
# programming language(C)

Eng\ Rana Mohamed Elbalka

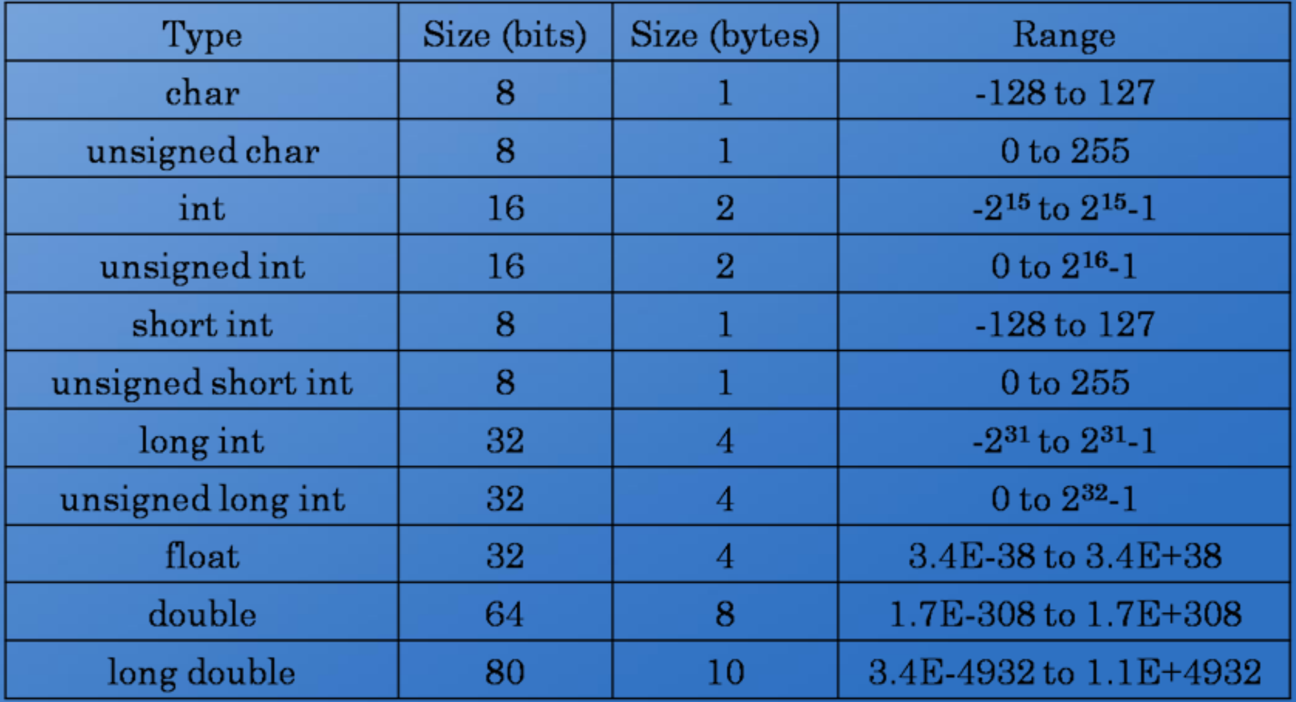
We will be using editor & compiler

Code::Blocks



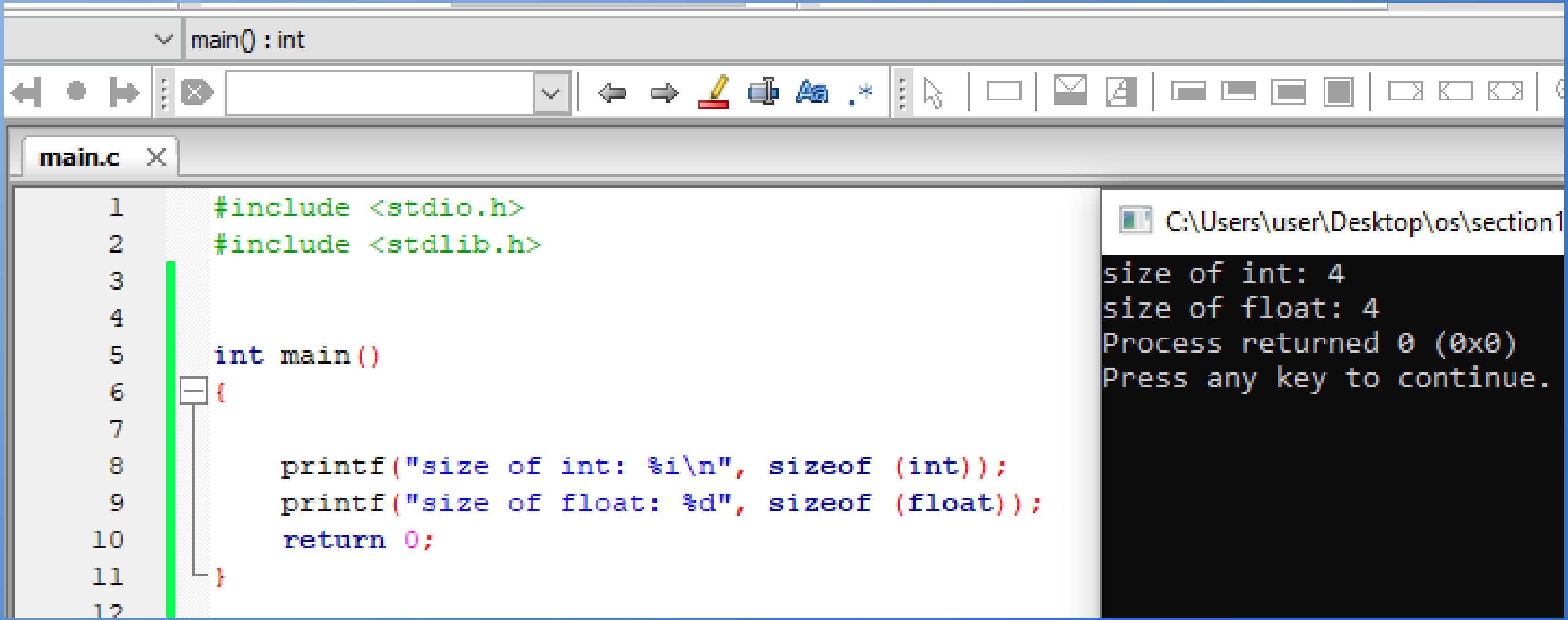
- Data types
  - Primitive type
    - char
    - int
    - float/ double
    - void
  - Derived type
    - signed
      - signed char
      - signed short
      - signed int
      - signed long
    - unsigned
      - unsigned char
      - unsigned short
      - unsigned int
      - unsigned long
    - long double
  - User defined type
    - Array
    - struct
    - union
    - enum

# Data type



| Type | Size (bits) | Size (bytes) | Range |
|---|---|---|---|
| char | 8 | 1 | -128 to 127 |
| unsigned char | 8 | 1 | 0 to 255 |
| int | 16 | 2 | $-2^{15}$ to $2^{15}-1$ |
| unsigned int | 16 | 2 | 0 to $2^{16}-1$ |
| short int | 8 | 1 | -128 to 127 |
| unsigned short int | 8 | 1 | 0 to 255 |
| long int | 32 | 4 | $-2^{31}$ to $2^{31}-1$ |
| unsigned long int | 32 | 4 | 0 to $2^{32}-1$ |
| float | 32 | 4 | 3.4E-38 to 3.4E+38 |
| double | 64 | 8 | 1.7E-308 to 1.7E+308 |
| long double | 80 | 10 | 3.4E-4932 to 1.1E+4932 |

# Example 1

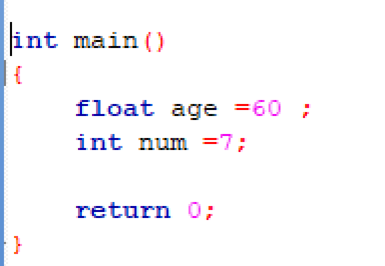


```c
#include <stdio.h>
#include <stdlib.h>


int main()
{

    printf("size of int: %i\n", sizeof (int));
    printf("size of float: %d", sizeof (float));
    return 0;
}
```

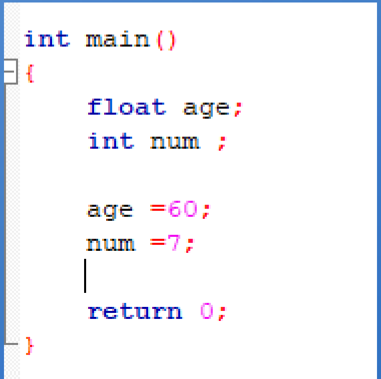
```
size of int: 4
size of float: 4
Process returned 0 (0x0)
Press any key to continue.
```

# variable

A variable is name given to a storage area that our programs can manipulate

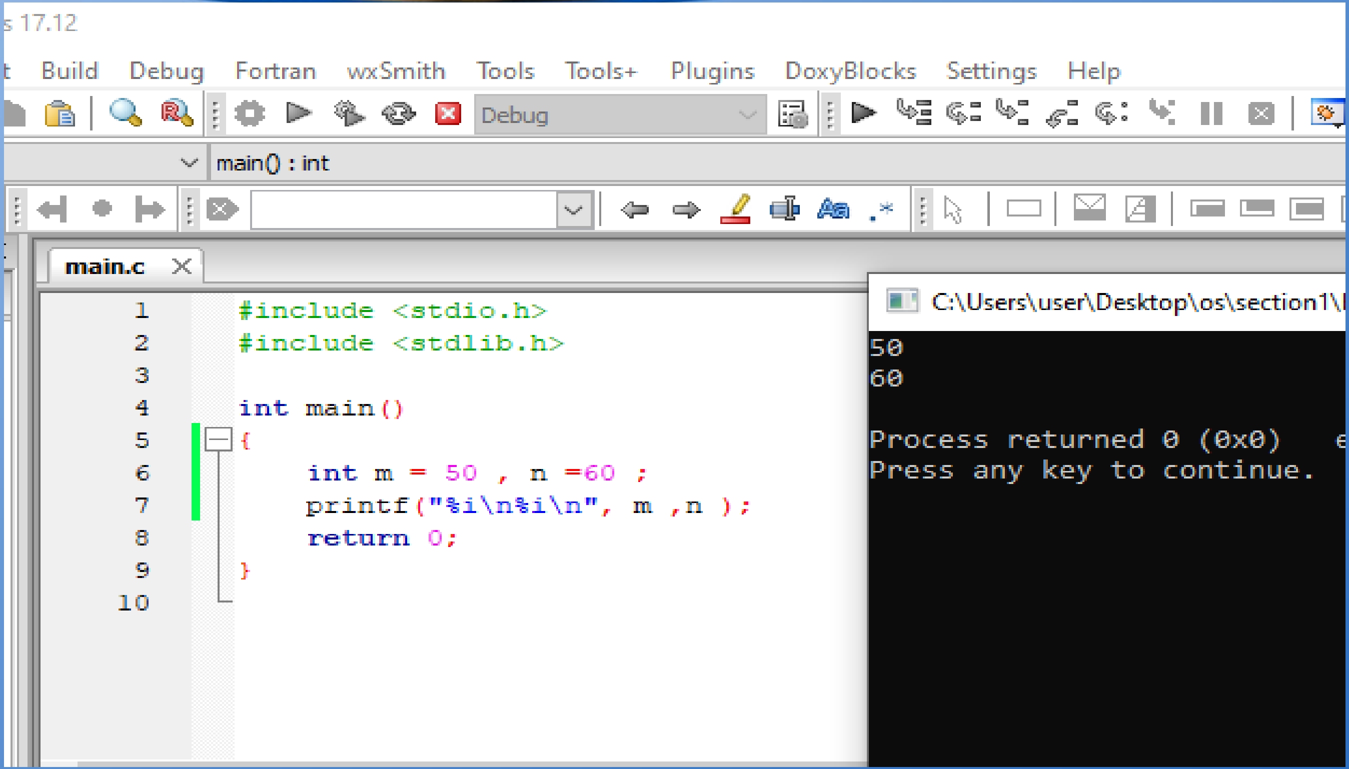


```
int main()
{
    float age =60 ;
    int num =7;

    return 0;
}
```



```
int main()
{
    float age;
    int num ;

    age =60;
    num =7;

    return 0;
}
```

# Example 2

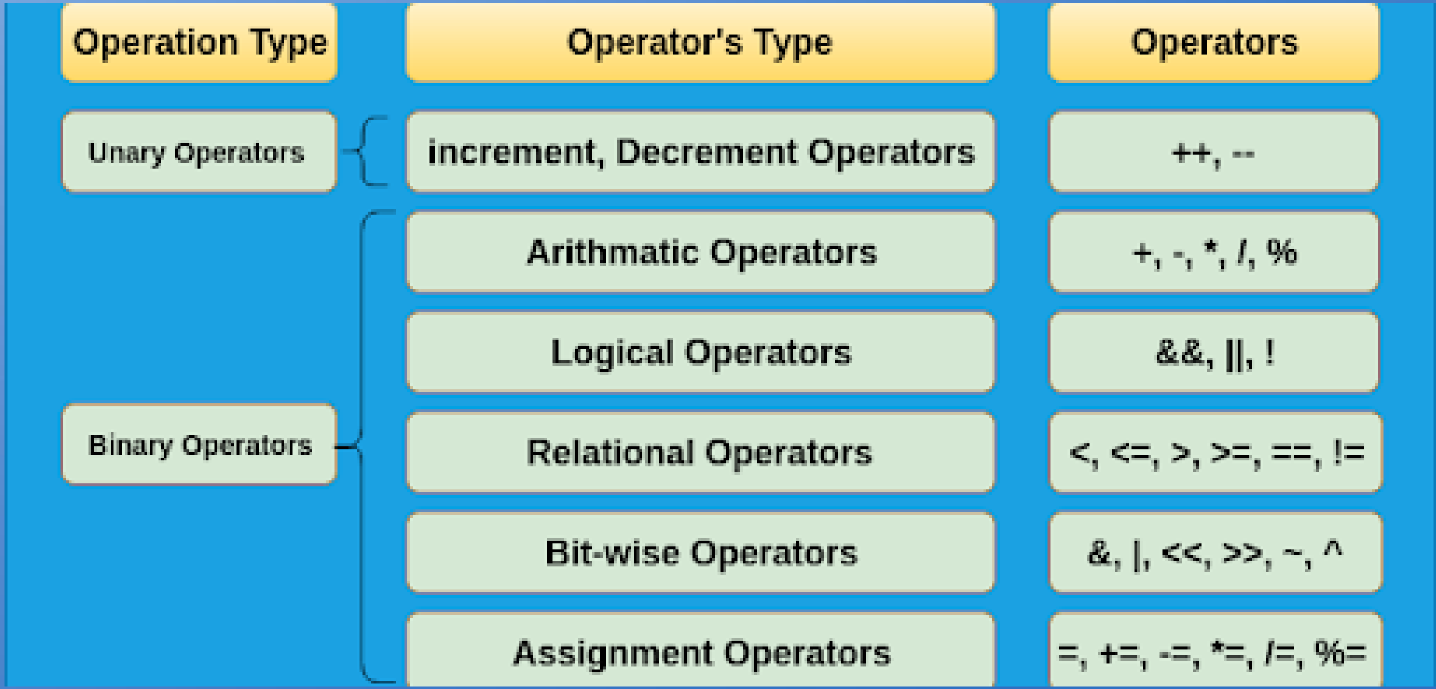


s 17.12
t Build Debug Fortran wxSmith Tools Tools+ Plugins DoxyBlocks Settings Help
Debug
main() : int
main.c
C:\Users\user\Desktop\os\section1\

```c
#include <stdio.h>
#include <stdlib.h>

int main()
{
    int m = 50 , n =60 ;
    printf("%i\n%i\n", m ,n );
    return 0;
}
```

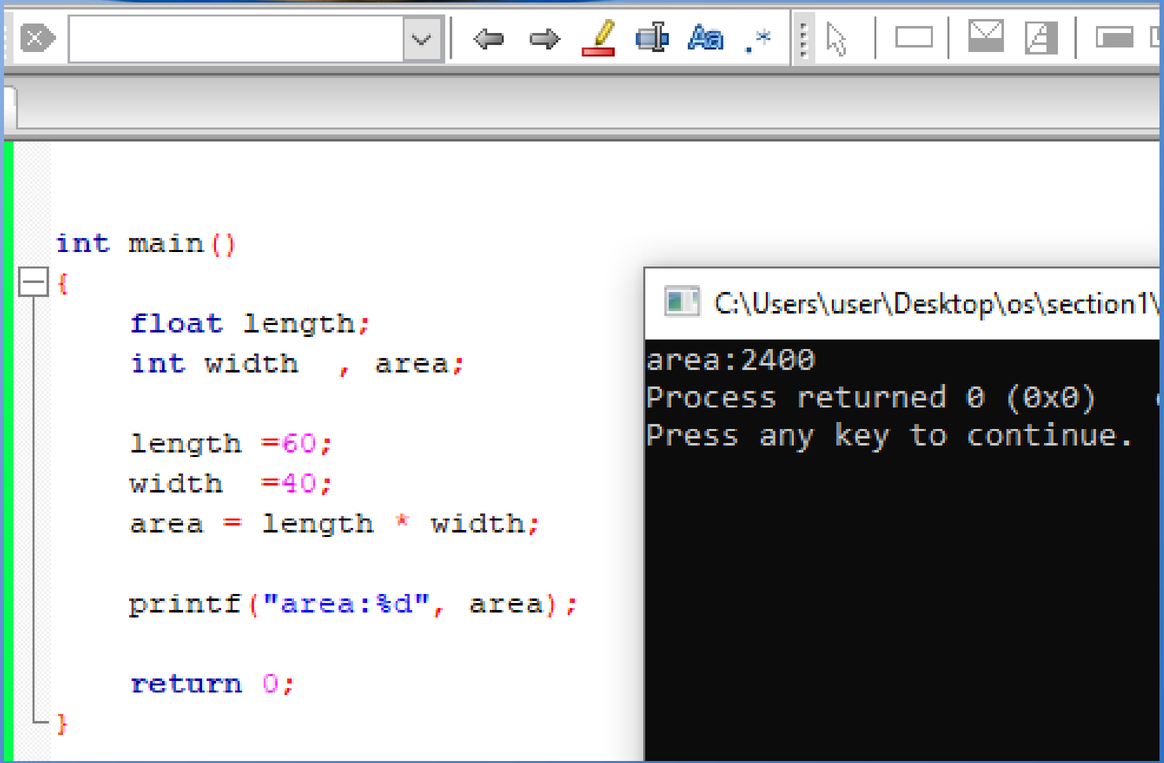
```
50
60

Process returned 0 (0x0)   e
Press any key to continue.
```

# Operators



| Operation Type | Operator's Type | Operators |
|---|---|---|
| Unary Operators | increment, Decrement Operators | ++, -- |
| Binary Operators | Arithmatic Operators | +, -, *, /, % |
| | Logical Operators | &&, \|\|, ! |
| | Relational Operators | <, <=, >, >=, ==, != |
| | Bit-wise Operators | &, \|, <<, >>, ~, ^ |
| | Assignment Operators | =, +=, -=, *=, /=, %= |

# Example 3

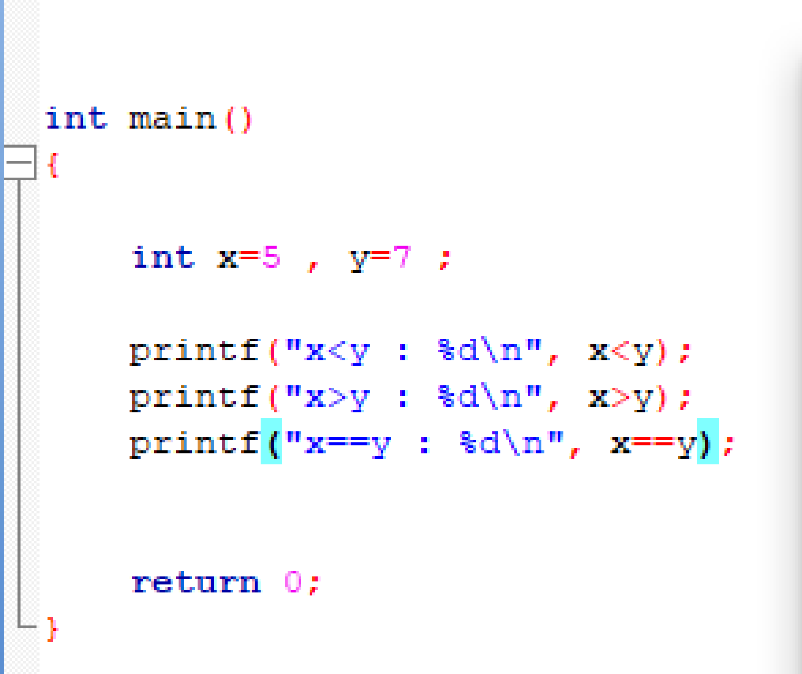


```
int main()
{
    float length;
    int width  , area;

    length =60;
    width  =40;
    area = length * width;

    printf("area:%d", area);

    return 0;
}
```

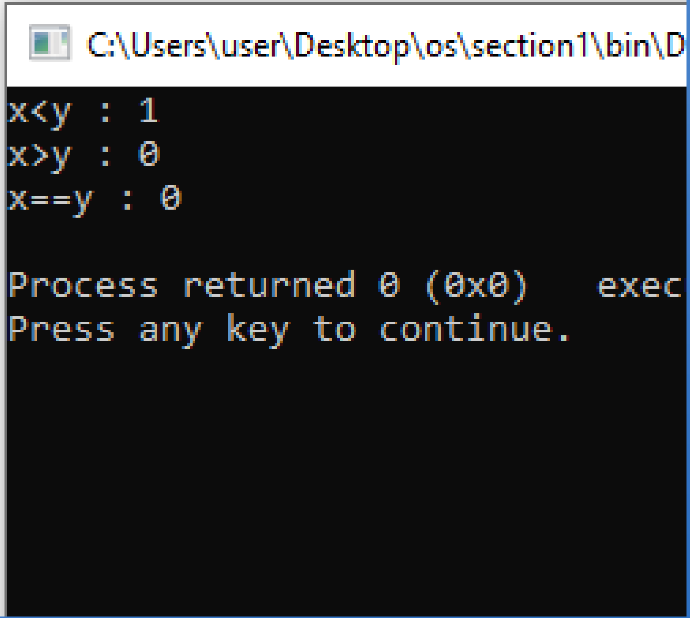
C:\Users\user\Desktop\os\section1\

```
area:2400
Process returned 0 (0x0)
Press any key to continue.
```

# Example 4

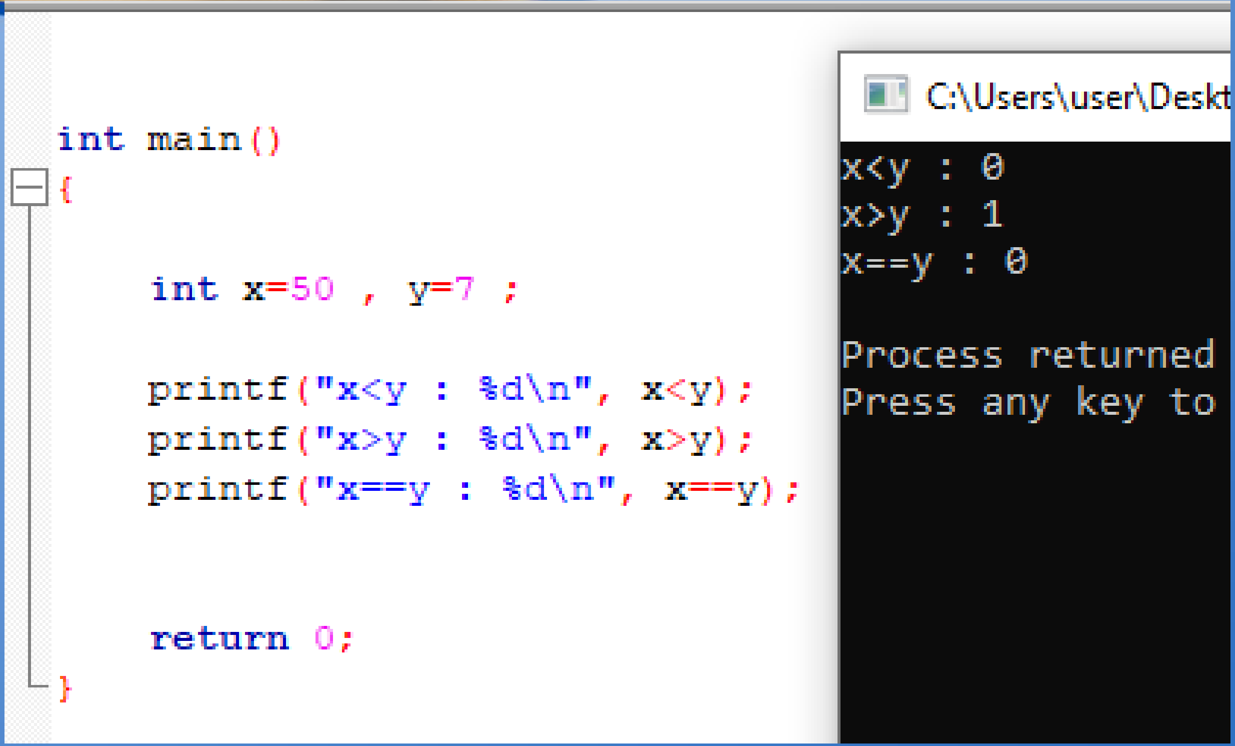


```
int main()
{

    int x=5 , y=7 ;

    printf("x<y : %d\n", x<y);
    printf("x>y : %d\n", x>y);
    printf("x==y : %d\n", x==y);


    return 0;
}
```



C:\Users\user\Desktop\os\section1\bin\D

```
x<y : 1
x>y : 0
x==y : 0

Process returned 0 (0x0)   exec
Press any key to continue.
```

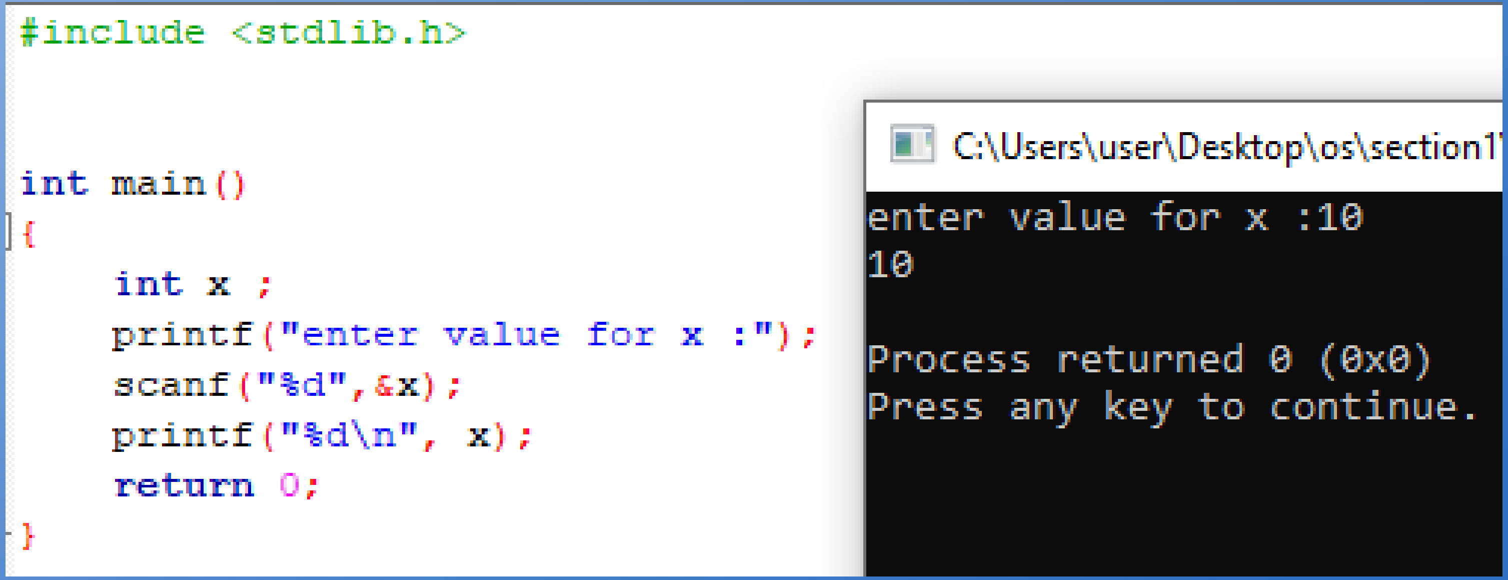


```
int main()
{

    int x=50 , y=7 ;

    printf("x<y : %d\n", x<y);
    printf("x>y : %d\n", x>y);
    printf("x==y : %d\n", x==y);


    return 0;
}
```

C:\Users\user\Deskt

```
x<y : 0
x>y : 1
x==y : 0

Process returned
Press any key to
```

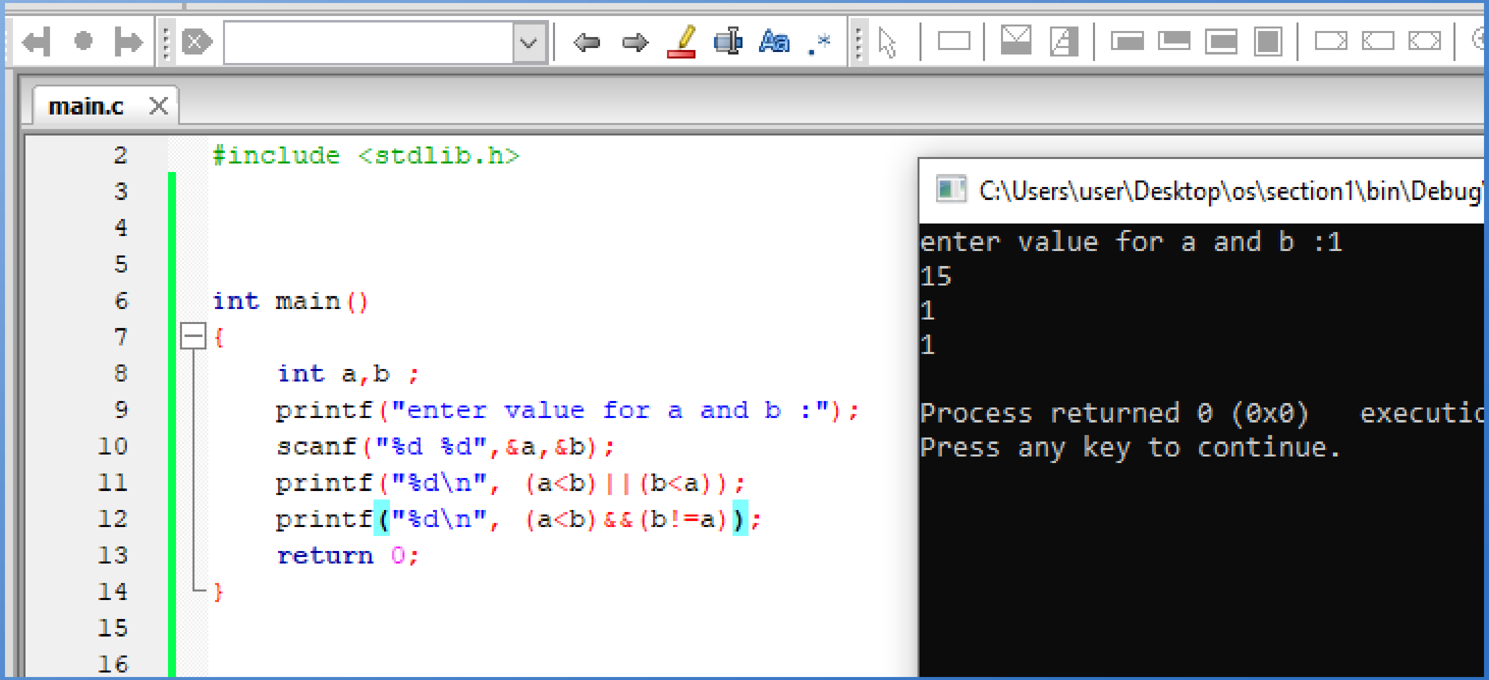
# scanf



```c
#include <stdlib.h>


int main()
{
    int x ;
    printf("enter value for x :");
    scanf("%d",&x);
    printf("%d\n", x);
    return 0;
}
```

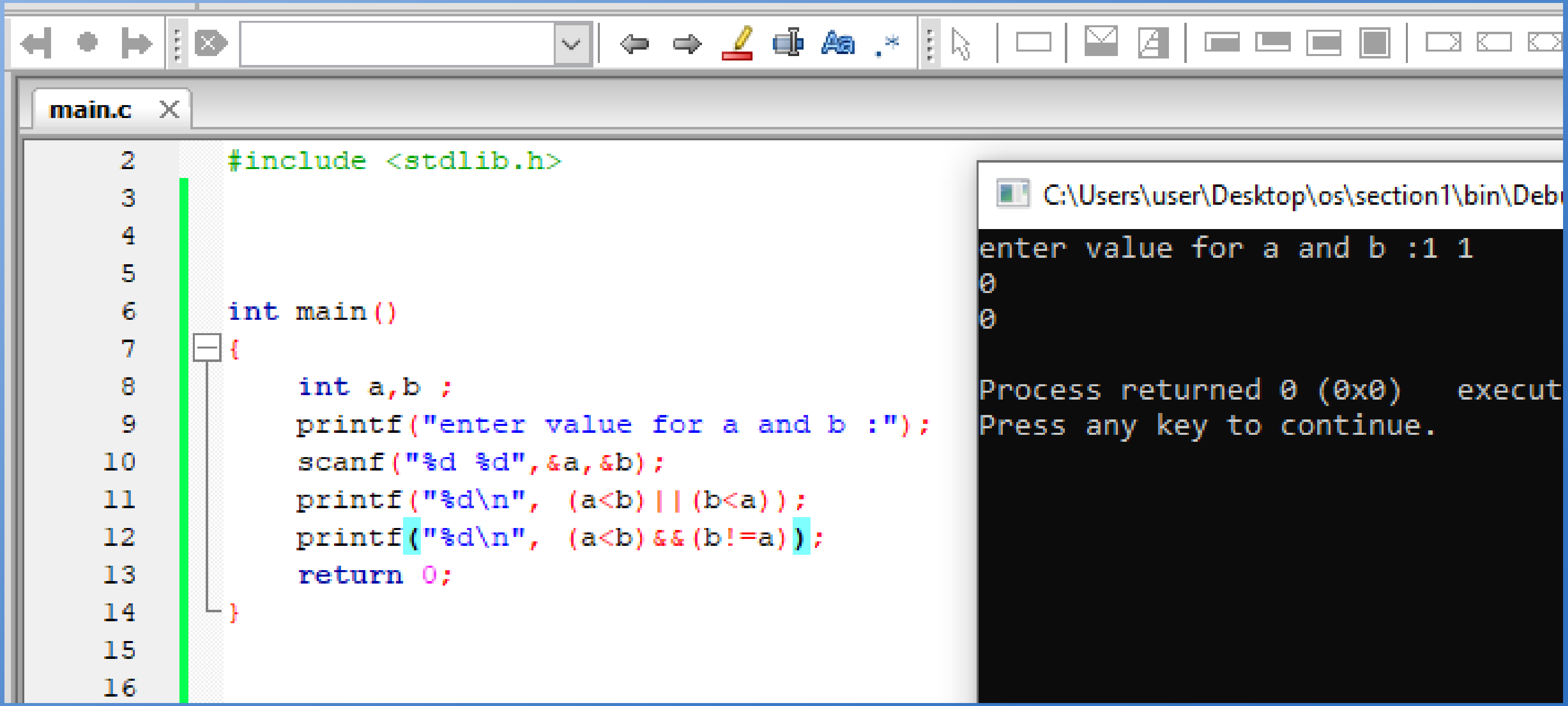
```
enter value for x :10
10

Process returned 0 (0x0)
Press any key to continue.
```

# Example 5



```c
#include <stdlib.h>



int main()
{
    int a,b ;
    printf("enter value for a and b :");
    scanf("%d %d",&a,&b);
    printf("%d\n", (a<b)||(b<a));
    printf("%d\n", (a<b)&&(b!=a));
    return 0;
}
```

```
enter value for a and b :1
15
1
1

Process returned 0 (0x0)   executic
Press any key to continue.
```



```c
#include <stdlib.h>



int main()
{
    int a,b ;
    printf("enter value for a and b :");
    scanf("%d %d",&a,&b);
    printf("%d\n", (a<b)||(b<a));
    printf("%d\n", (a<b)&&(b!=a));
    return 0;
}
```

C:\Users\user\Desktop\os\section1\bin\Debu

```
enter value for a and b :1 1
0
0

Process returned 0 (0x0)   execut
Press any key to continue.
```

# Thank You