

# Llamadas por referencia

Programación estructurada

# Llamadas por valor y por referencia

Existen dos formas de pasar valores a una función.

- Llamada por valor (En C solo se realizan llamadas por valor)
- Llamada por referencia (En C se pueden simular llamadas por referencia)

# Llamadas por valor

```
#include <stdio.h>
#include <conio.h>
```

```
int cubo_por_valor(int);
```

```
main ()
{
    int num=5;
    printf("El valor original de num es %d\n",num);
    num=cubo_por_valor(num);
    printf("El nuevo valor de num es %d\n",num);
    getch ();
    return 0;
}
```

```
int cubo_por_valor(int n)
{
    return n*n*n;
}
```

# Llamadas por referencia

```
#include <stdio.h>
#include <conio.h>
```

```
void cubo_por_referencia(int *);
```

```
main ()
{
    int num=5;
    printf("El valor original de num es %d\n",num);
    cubo_por_referencia(&num);
    printf("El nuevo valor de num es %d\n",num);
    getch ();
    return 0;
}
```

```
void cubo_por_referencia(int *nPtr)
{
    *nPtr=*nPtr**nPtr**nPtr;
}
```

Antes de que main llame a cubo\_por\_valor

```
main()
int number = 5;
number = cubo_por_valor (number)
```

**number = 5**

```
Int cubo_por_valor(int n)
Return n*n*n;
```

**n = indefinido**

Después de que cubo\_por\_valor recibe la llamada

```
main()
int number = 5;
number = cubo_por_valor (number)
```

**number = 5**

```
Int cubo_por_valor(int n)
Return n*n*n;
```

**n = 5**

Después de que cubo\_por\_valor eleva al cubo a n

```
main()
int number = 5;
number = cubo_por_valor (number)
```

**number = 5**

```
Int cubo_por_valor(int n)
Return n*n*n;
```

**125** **n = 5**

Después de que cubo\_por\_valor regresa a main

```
main()
int number = 5;
number = cubo_por_valor (number)
```

**number = 5**

```
Int cubo_por_valor(int n)
Return n*n*n;
```

**n = indefinido**

Después de que main termina la asignación a number

```
main()
int number = 5;
number = cubo_por_valor (number)
```

**number = 125**

```
Int cubo_por_valor(int n)
Return n*n*n;
```

**n = indefinido**

Antes de la llamada por referencia a cubo\_x\_referencia

```
main()
int number = 5;
number=cubo_x_referencia (&number)
```

**number = 5**

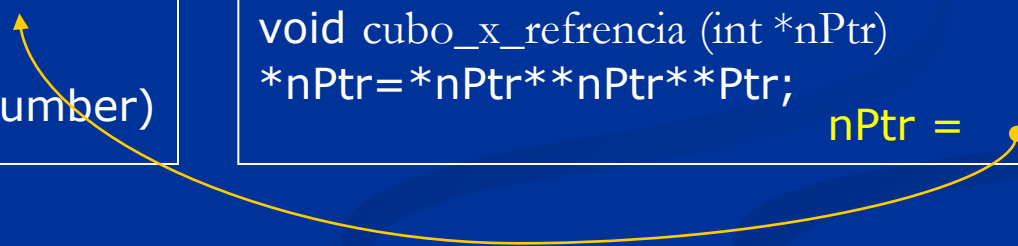
```
void cubo_x_referencia (int *nPtr)
*nPtr=*nPtr**nPtr**Ptr;
nPtr = indefinido
```

Después de la llamada a cubo\_x\_referencia y antes de que \*nPtr se eleve al cubo

```
main()
int number = 5;
number=cubo_x_referencia (&number)
```

**number = 5**

```
void cubo_x_referencia (int *nPtr)
*nPtr=*nPtr**nPtr**Ptr;
nPtr =
```



Después de que \*nPtr es elevado al cubo

```
main()
int number = 5;
number=cubo_x_referencia (&number)
```

**number = 125**

```
void cubo_x_referencia (int *nPtr)
*nPtr=*nPtr**nPtr**Ptr;
nPtr =
```

