

Funciones y procedimientos: Tipos pasos de
Parámetros de una función.
Generación de números aleatorios.

Fundamentos de programación

Agenda

- Tipos de pasos de parámetro de una función.
- Generación de números aleatorios.

Pasos de parámetros

- Cuando una función recibe parámetros de entrada, ésta realiza una copia local de cada parámetro.

Pasos de parámetros

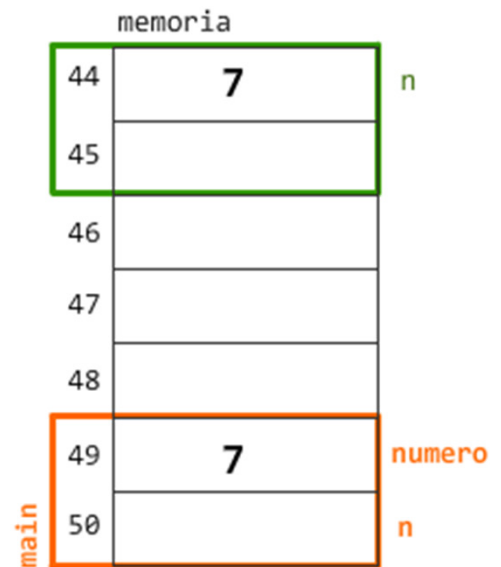
Paso por valor

- Cuando una función recibe parámetros de entrada, ésta realiza una copia local de cada parámetro.
- Este tipo de paso de parámetros de los denomina “paso por valor”

```
#include <stdio.h>

int cuadrado(int n)
{
    int cuad;
    cuad = n*n;
    return cuad;
}

int main()
{
    int numero, n;
    printf("Ingrese un numero: ");
    scanf("%i", &numero);
    n = cuadrado(numero);
    printf("El cuadrado de %i es %i", numero,n);
}
```



Ingrese un numero: 7

PANTALLA

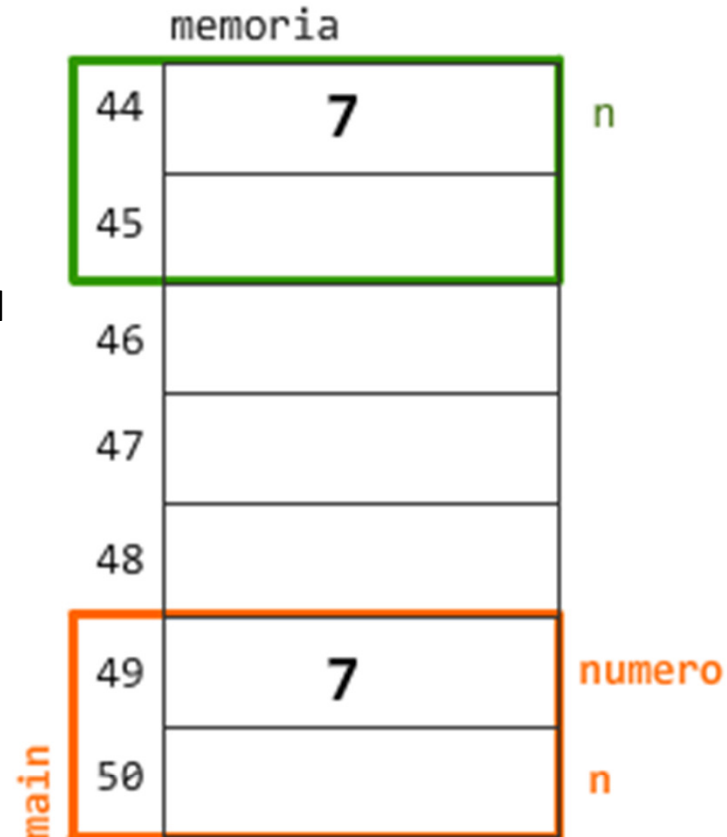
Pasos de parámetros: Paso por valor

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

```
int cuadrado(int n)  
{  
    int cuad;  
    cuad = n*n;  
    return cuad;  
}
```

```
int main()  
{  
    int numero, n;  
    printf("Ingrese un numero: ");  
    scanf("%i", &numero);  
    n = cuadrado(numero);  
    printf("El cuadrado de %i es %i", numero,n);  
}
```

- Los cambios hechos a la copia no afectan al valor original.
- Usado cuando no se necesita modificar el argumento
 - Evita modificaciones accidentales



```
Ingrese un numero: 7
```

PANTALLA

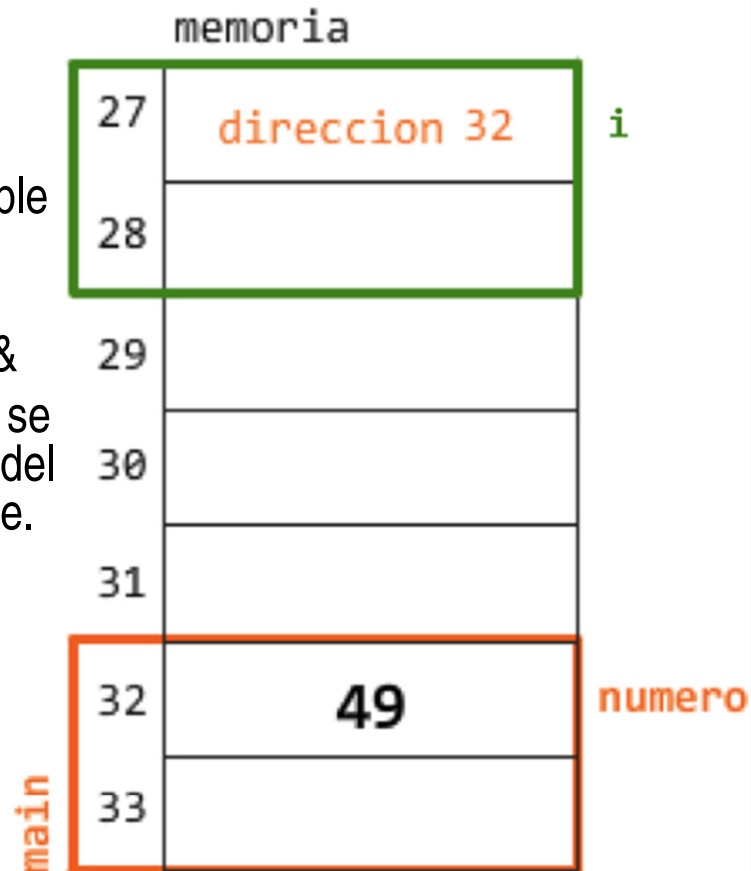
Pasos de parámetros: por referencia

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

```
void cuadrado(int *i)
{
    *i=(*i)*(*i);
}
```

```
int main()
{
    int numero;
    printf("Ingrese un numero: ");
    scanf("%i", &numero);
    cuadrado(&numero);
    printf("El cuadrado es %i", numero);
}
```

- La función recibe la dirección de la variable como parámetro
- La variable se envía acompañada de un &
- Dentro de la función se agrega un * al inicio del nombre de la variable.



```
Ingrese un numero: 7
```

PANTALLA

Paso por referencia

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

```
void cuadrado(int *i)
{
    *i=(*i)*(*i);
}
```

```
int main()
{
    int numero;
    printf("Ingrese un numero: ");
    scanf("%i", &numero);
    cuadrado(&numero);
    printf("El cuadrado es %i", numero);
}
```

- Pasa el argumento original
- Cambios en la función modifican a los originales
- Sólo usado con ciertas funciones



```
Ingrese un numero: 7
```

PANTALLA

Pasos de parámetros

Paso por referencia

- Los arreglos siempre se pasan por referencia.
- Ejemplo: Implemente el método `zeros(int arreglo[], int tam)`, que recibe un arreglo de tamaño `tam` y lo llena de ceros.

Generación de números aleatorios

Función rand <stdlib.h>

- Retorna un valor “aleatorio” entre 0 y RAND_MAX (mínimo 32767)

```
i=rand();
```

- Factor de escalamiento

– Obtener un valor entre 1 y N

```
1 + (rand () % N)
```

rand() % N : retorna un valor entre 0 y N-1

Ejemplo: un valor aleatorio entre 1 y 6

```
1 + (rand ( ) % 6)
```

Generación de números aleatorios

Función `srand` `<stdlib.h>`

- `Rand` produce un misma secuencia (pseudoaleatorio).
- Podemos iniciar la secuencia donde queramos.
- `srand` toma como parámetro un entero y “brinca” a ese número la secuencia de números aleatorios
 - `srand(entero)`
- Ejemplo: `srand(time(NULL))`
 - `time.h`
 - `Time(N)`: retorna el tiempo (segundos) en el cual el programa fue compilado

Ejercicio

- Una empresa de casinos desea realizar un juego de dados que permita a los usuarios jugar mediante apuestas definidas. Al inicio del juego el usuario indica su apuesta para luego dar inicio el juego. Para esto el usuario debe seleccionar el tipo de juego que quiere realizar:
 - Par. Donde apuesta que el par de dados sumaran un número par. Para lo cual gana el 25% de lo apostado.
 - Impar. Donde apuesta que el par de dados sumaran un número impar. Para lo cual gana el 25% de lo apostado.
 - Número fijo. Donde apuesta que el par de dados sumaran un número definido por el usuario. Para lo cual gana el 100% de lo apostado.

En esta clase Ud. aprendió

- Identificar las diferentes maneras como podemos enviar parámetros a las funciones.
- Seleccionar la manera más adecuada de enviar parámetros a las funciones.
- Generar números aleatorios en C.
- Identificar los casos en los cuales es indispensable generar números aleatorios.