

---

# Practica de Funciones

# Ejemplo de uso de funciones

```
/* **** */
* Programa: factorial.c *
* Descripción: Calcula el factorial de un numero usando una funcion *
* con tipo de dato long y double *
* Autor: Pedro Corcuera *
* Revisión: 1.0 2/02/2008 *
\ **** /
#include <stdio.h>
```

```
long int factoriali (int n);
double factorialf (int n);
```

declaración de prototipos

```
int main (void)
{
    int n;
    long int fact;

    printf ("Ingresar numero: ");
    scanf("%d", &n);
    fact = factoriali(n);
    printf("El factorial de %d es: %d\n",n, fact);
    printf("El factorial de %d es: %.0f\n",n, factorialf(n));
}
```

# Ejemplo de uso de funciones

---

```
long int factoriali ( int n )  
{  
    int i;  
    long int producto;  
  
    producto = 1;  
    for ( i = n; i > 1; i-- ) {  
        producto *=i;  
    }  
    return (producto);  
}
```

variables locales

```
double factorialf ( int n )  
{  
    int i;  
    double producto;  
  
    producto = 1;  
    for ( i = n; i > 1; i-- ) {  
        producto *=i;  
    }  
    return (producto);  
}
```

# Torres de hanoi: programa

---

```
/******\
* Programa: thanoi.c *
* Descripción: Programa que imprime los pasos para resolver el juego de *
*               las torres de hanoi para n discos dados por el usuario *
* Autor: Pedro Corcuera *
* Revisión: 1.0 2/02/2008 *
\*****/
#include <stdio.h>
#define NUM_MAX_DISCOS 16

void torres_hanoi(int n, char detorre, char atorre, char aux torre);

main(void)
{
    int n;

    printf("Torres de Hanoi: Cuantos discos? (max. %d) ", NUM_MAX_DISCOS);
    scanf("%d", &n);
    if (n > NUM_MAX_DISCOS)
    {
        printf("Numero de discos muy grande\n");
        return 1;
    }
    torres_hanoi(n, 'A', 'C', 'B');
    return 0;
}
```

---

# Torres de hanoi: función recursiva

---

```
/******\
* Funcion: torres_hanoi *
* Descripción: imprime los pasos para resolver el juego de las torres de *
* hanoi para n discos de manera recursiva *
* Argumentos: int n: numero de discos *
* char detorre: caracter del pivote donde estan los discos *
* char atorre: caracter del pivote destino *
* char aux torre: caracter del pivote auxiliar *
* Valor devuelto: ninguno (la funcion imprime los movimientos) *
* Autor: Pedro Corcuera *
* Revisión: 1.0 2/02/2008 *
\*****/
void torres_hanoi(int n, char detorre, char atorre, char aux torre)
{
    if (n > 0)
    {
        torres_hanoi(n-1, detorre, aux torre, atorre);
        printf("mover disco %d de torre %c a torre %c\n", n, detorre, atorre);
        torres_hanoi(n-1, aux torre, atorre, detorre);
    }
    return;
}
```

# Cambio de base: programa

---

```
/* **** */
* Programa: cambase.c *
* Descripción: Programa que imprime el numero N en Base 2 <= Base <= 16 *
* Autor: Pedro Corcuera *
* Revisión: 1.0 2/02/2008 *
\ **** /
#include <stdio.h>

void cambia_base( unsigned int N, unsigned int Base);

main()
{
    int i;

    i = 51;
    printf("%d Base 10, ", i);
    printf("\t Base 2: ");
    cambia_base(i, 2);
    printf("\t Base 8: ");
    cambia_base(i, 8);
    printf("\t Base 16: ");
    cambia_base(i, 16);
    printf("\n");
}
```

---

# Cambio de base: función

---

```
/******\
* Funcion: cambia_base *
* Descripción: imprime un numero entero N en la base Base 2 <= Base <= 16*
* de manera recursiva *
* Argumentos: int N: numero decimal *
* int Base: base a la que se desea convertir 2 <= Base <= 16 *
* Valor devuelto: ninguno (la funcion imprime el numero) *
* Autor: Pedro Corcuera *
* Revisión: 1.0 2/02/2008 *
\*****/
void cambia_base( unsigned int N, unsigned int Base )
{
    static char Tabla_Digitos[ ] = "0123456789abcdef";

    if( N >= Base )
        cambia_base( N / Base, Base );
    putchar( Tabla_Digitos[ N % Base ] );
}
```