

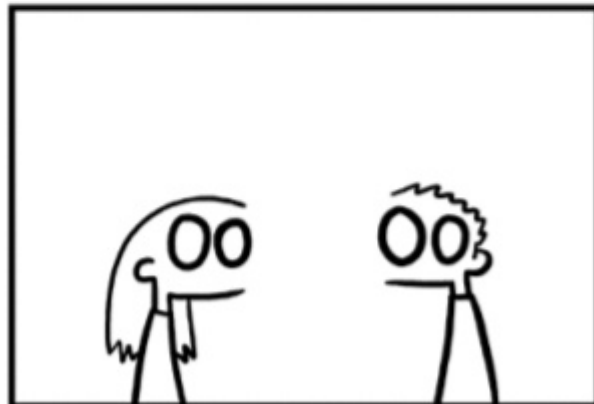
FUNCIONES RECURSIVAS

FUNCIONES RECURSIVAS

Es una funcion que se llama a si misma.







FUNCIONES RECURSIVAS

Se requieren 3 cosas

- Una Función
- Un Caso Base
- Un Caso Recursivo

EJERCICIO

Escribir un programa que calcule el factorial de un número proporcionado por el usuario

ejemplo:

$5! = 5 * 4 * 3 * 2 * 1$

EJERCICIO

```
//Paso 1: definir la función
```

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

```
long factorial( int n );
```

```
int main (){
```

```
}
```

```
long factorial( int n ){
```

```
}
```


EJERCICIO

```
//Paso 2: Caso base
```

ejemplo:

$$5! = 5 * 4 * 3 * 2 * 1$$

EJERCICIO

//Paso 2: Caso base

ejemplo:

$$5! = 5 * 4 * 3 * 2 * 1$$

$$5! = 5 * 4!$$

$$4! = 4 * 3!$$

$$3! = 3 * 2!$$

$$2! = 2 * 1!$$

$$1! = 1$$

EJERCICIO

```
//Paso 2: Caso base

#include <stdio.h>

long factorial( int n );

int main (){

}

long factorial( int n ){
    if (n == 1 ){
        return 1;
    }else{

    }

}
```

EJERCICIO

```
//Paso 3: Funcion recursiva

#include <stdio.h>

long factorial( int n );

int main (){

}

long factorial( int n ){
    if (n == 1 ){
        return 1;
    }else{
        return n * factorial(n-1);
    }
}
```

EJERCICIO

```
int main (){

    int numero;
    long total;

    printf("\n Proporcione un numero: ");
    scanf("%i", &numero);

    total = factorial(numero);

    printf("\n El factorial de %i es %li ", numero, total);

}
```

EJERCICIO

Escriba un programa que calcule la serie de fibonacci de un número proporcionado por el usuario

Nota: resolver utilizando recursión

ejemplo:

número = 10

serie = 0 1 1 2 3 5 8 13 21 34