

Fundamentos de programación

Funciones recursivas

Facultad de Ingeniería / Escuela de Informática
Universidad Andrés Bello, Viña del Mar.

- Caso base: son casos simples, si la función es llamada con el caso base retorna un resultado.
- Caso general: es un caso más complejo, la función se llama a sí misma con otro argumento de entrada.

Ejemplo: factorial

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

Ejercicio 1

Defina una función recursiva que recibe un número n e imprima los números del n al 1

Defina una función recursiva que permita calcular el n -ésimo elemento de la serie de fibonacci, que corresponde a

0, 1, 1, 2, 3, 5, 8, 13, 21, ...

Puede ser definido de forma recursiva por

- $\text{fibonacci}(0)=0$
- $\text{fibonacci}(1) = 1$
- $\text{fibonacci}(n) = \text{fibonacci}(n-1)+\text{fibonacci}(n-2)$

Ejercicio 3

Defina una función que permita calcular los elementos del triángulo de pascal según se define a continuación.

Lo utilizamos para calcular las C_m^n . Este número cuenta la cantidad de formas de escoger m objetos de un total de n objetos distintos. Les llamamos *combinaciones* de m objetos en n .

El triángulo de Pascal se construye como sigue: al principio se coloca un 1 (que corresponde a C_0^0). Para cada renglón subsecuente, digamos para el renglón n , se coloca un 1 a la izquierda y un 1 a la derecha que corresponden con C_0^n y C_n^n respectivamente. Los elementos restantes se calculan sumando los dos números que tiene justo arriba a la izquierda y a la derecha, es decir $C_m^n = C_{m-1}^{n-1} + C_m^{n-1}$ para todo $0 < m < n$.

| | | | | |
|---|---|---|---|---|
| | | 1 | | |
| | 1 | | 1 | |
| | 1 | 2 | 1 | |
| 1 | 3 | 3 | 1 | |
| 1 | 4 | 6 | 4 | 1 |

Ejercicio 4

Defina una función recursiva que permita imprimir un número como string de caracteres.