

Práctica 17. Funciones recursivas

1. Introducción

Las funciones recursivas son aquellas que se llaman a sí mismas. Una función recursiva tiene dos partes:

- Caso trivial: sería la solución inmediata, que devuelve la función
- Caso recursivo: sería la parte de la función que se llama a sí misma para calcular el resultado

Como en el lenguaje ensamblador al llamar a otra función se pierde el valor del registro `ra`, hay que apilarlo en la pila previamente. La estructura de la función será:

```
1 Caso trivial:
2   Se calcula el resultado
3   Se devuelve el resultado en a0
4
5 Caso recursivo:
6   Se apila ra
7   Se apila lo que se necesite (a0, etc)
8
9   Se pasan los argumentos en a0, a1, etc
10  jal ra, funcion
11  Se cogen los resultados de a0 para calcular el resultado final (
    se desapila a0, etc o lo que se necesite)
12  Se devuelve el resultado en a0
13
14  Se desapila ra
15  ret
```

2. Ejercicios con funciones recursivas

2.1. Ejercicio

Realice un programa que solicite al usuario un número entero n y que calcule el factorial de dicho número, haciendo uso de una subrutina recursiva llamada `factorial`. El argumento que se debe pasar es el número indicado por el usuario.

2.2. Ejercicio

Realice un programa que solicite al usuario un número entero n y que calcule el número fibonacci que corresponde, haciendo uso de una subrutina recursiva llamada `fibonacci`. El argumento que se debe pasar es el número indicado por el usuario.

3. Otros ejercicios

3.1. Ejercicio

Escriba un programa que solicite al usuario una cadena de texto y que posteriormente la copie en otra cadena imprimiendo el resultado por pantalla. Se le debe pasar a la subrutina `strcpy` dos argumentos: un puntero a la cadena original y un puntero a la cadena destino. NOTA: Recordad que las cadenas terminan con el carácter `'\0'` que tiene el valor 0.

3.2. Ejercicio

Realice un programa que solicite al usuario dos números enteros a y b y llame a una función que diga cuántos números pares y cuántos impares hay entre a y b , devolviendo ese resultado. Los argumentos que se deben pasar son los números indicados por el usuario. La función `main` imprimirá los resultados por pantalla.

3.3. Ejercicio

Escriba un programa que solicite al usuario una cadena de texto y una letra. La función `main` pasa como argumentos a una función la cadena leída, la letra leída y una cadena destino. Esa función copia en la cadena destino la cadena

leída salvo los caracteres que coincidan con la letra leída y devuelve cuántas letras ha eliminado de la cadena original (cuántas letras iguales a la letra leída ha dejado de escribir). La función `main` imprimirá por pantalla la cadena original leída, el número de letras eliminadas y la cadena resultante. NOTA: Recordad que las cadenas terminan con el carácter `'\0'` que tiene el valor 0.

3.4. Ejercicio

Realice un programa que lea una cadena de máximo 100 caracteres y dos números positivos. Si no son positivos, dará un mensaje de error y volverá a leerlos hasta que lo sean. Llamará a una función que le pase la cadena y los dos enteros leídos, la cual copiará en una cadena destino, también pasada como argumento la subcadena que existe de la posición indicada por el número menor hasta la posición indicada por el número mayor. La función `main` imprimirá la cadena resultante por pantalla. Por ejemplo, si lee la cadena `Esto es una cadena` y los números 2 y 7 imprimirá la cadena `sto es`.

3.5. Ejercicio

Considérese el siguiente fragmento de código que representa el segmento de datos:

```
1      .data
2      .byte 4
3      .align 1
4      .half 5
5      .align 2
6      .word 6
7      .byte 'b'
8 datos: .byte 7
```

Escriba un programa que, utilizando los modos de direccionamiento realice las siguientes operaciones sin modificar el segmento de datos:

- Calcular la suma del `half` que vale 5 y del `word` que tiene valor 6 y colocar este resultado en el `byte` que vale 7 bytes. Imprimir y comprobar el valor.
- Utilizando la etiqueta `datos`, almacenar el carácter ASCII `'a'` en el byte inicializado a 4. Imprimir y comprobar el valor.