



## Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

# Laboratorios de computación salas A y B

*Profesor:* Adrian Ulises Mercado Martinez

*Asignatura:* Estructura de Datos y Algoritmos I

*Grupo:* 13

*No de Práctica(s):* 12

*Integrante(s):* Monroy Salazar Diego Gustavo

*No. de Equipo de  
cómputo empleado:*

*No. de Lista o Brigada:* Brigada 9

*Semestre:* 2020-2

*Fecha de entrega:* 07/06/2020

*Observaciones:*

**CALIFICACIÓN:** \_\_\_\_\_

# Introducción

La recursividad trata de que una función se llame a si misma de tal manera que resuelva un problema de manera “natural”, este método es útil cuando encontrar una solución iterativa a un problema se torna complicado, aunque suele usar más recursos.

## Desarrollo

1. Adaptamos una función que hicimos previamente para borrar los nodos de una lista de tal forma que lo haga recursivamente.

```
1  #include <stdio.h>
2  #include "e1.h"
3  #include <string.h>
4
5  int main(){
6      LIST *lista;
7      INFO info;
8      strcpy(info.nombre, "nombre1");
9      strcpy(info.apellido, "apellido11 apellido 12");
10     lista=crear_lista();
11     insertar(info, lista);
12     imprimir(lista);
13     eliminar(lista);
14     return 0;
15 }
```

---

```
void borrar_nodos(NODE *n)
{
    if(n->next!=NULL){ //caso recursivo
        borrar_nodos(n->next);
    }
    n->prev=NULL; //caso base
    free(n);
}
```

2. Creamos una función iterativa que imprime huellas de una tortuga de tal manera que forme una espiral.

```
"""
Funcion iterativa
"""

import turtle
wn=turtle.Screen()
wn.bgcolor("lightgreen")
wn.title("Tortuga")
tess=turtle.Turtle()
tess.shape("turtle")
tess.color("blue")
tess.penup()
size=20
for i in range(30):
    tess.stamp()
    size=size+3
    tess.forward(size)
    tess.right(24)

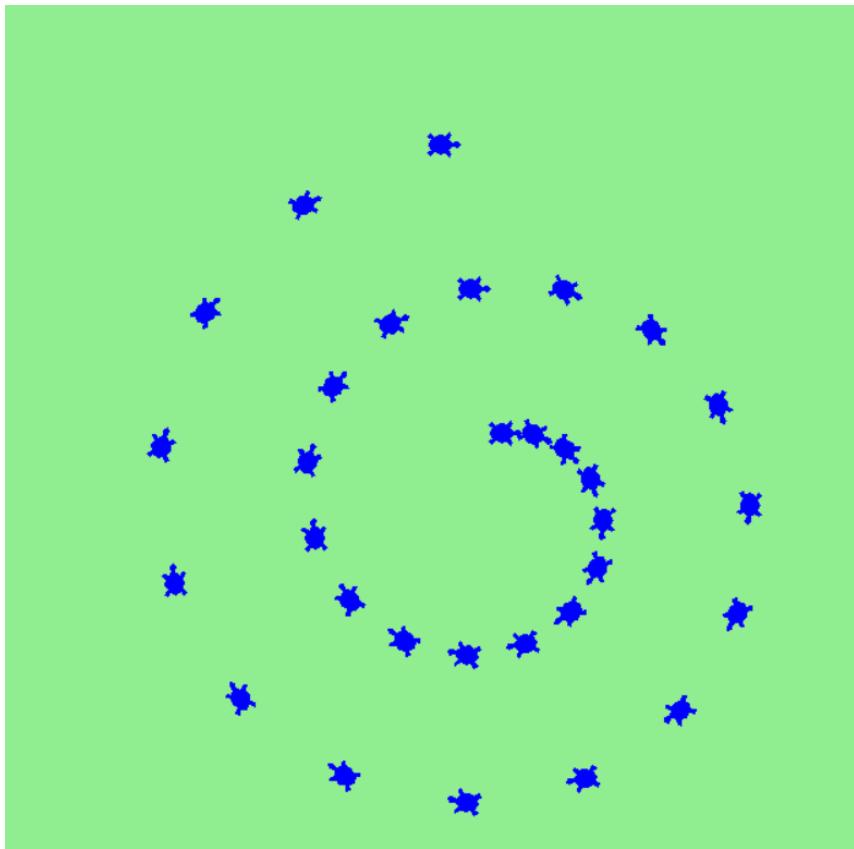
wn.mainloop()
```

3. Adaptamos la función anterior de tal forma que se haga recursiva, de esta manera le vamos pasando datos que formaran el camino de la tortuga a partir de la operación base que es dibujar una huella.

```
"""
Funcion recursiva
"""

import turtle
import argparse
def recorrido_recursivo(tortuga, espacio, huella):
    if huella>0:
        tortuga.stamp()
        espacio=espacio+3
        tortuga.forward(espacio)
        tortuga.right(24)
        recorrido_recursivo(tortuga, espacio, huella-1)

ap=argparse.ArgumentParser()
ap.add_argument("--huella", required=True, help="numero de huellas")
args=vars(ap.parse_args())
huellas=int(args["huella"])
wn=turtle.Screen()
wn.bgcolor("lightgreen")
wn.title("Tortuga")
tess=turtle.Turtle()
tess.shape("turtle")
tess.color("blue")
tess.penup()
recorrido_recursivo(tess, 20, 30)
wn.mainloop()
```



## Conclusión

La recursividad es una herramienta más que nos ayudara a resolver un problema de manera sencilla, aunque no es siempre conveniente por la sencillez de algunos problemas que se pueden resolver de forma iterativa y porque suele consumir más recursos.