



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Profesor: Adrian Ulises Mercado

Asignatura: Estructura de datos y algoritmos I

Grupo: 13

No de Práctica(s): 12

Integrante(s): Pacheco Salgado Mauricio

*No. de Equipo de
cómputo empleado:*

No. de Lista o Brigada: 10

Semestre: 2020-1

Fecha de entrega:

Observaciones:

CALIFICACIÓN: _____

INTRODUCCION.

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

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

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include "E1.h"

int main(){
    LIST *lista;
    INFO info;
    strcpy(info.nombre, "nombre1");
    strcpy(info.apellido, "apellido 11 apellido12");

    lista = crear_lista();
    insertar(info, lista);
    strcpy(info.nombre, "nombre2");
    strcpy(info.apellido, "apellido 21 apellido22");

    insertar(info, lista);
    imprimir(lista);
    strcpy(info.nombre, "nombre3");
    strcpy(info.apellido, "apellido 31 apellido23");

    insertar(info, lista);
    imprimir(lista);
    eliminar(lista);
    return 0;
}
```

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

2. E4.py

Utilizamos un programa de una función iterativa lo que hace que imprima en un diseño huellas de una tortuga en una forma de espiral, el fondo y el color nosotros se lo brindamos, pero en este caso es verde

```
import turtle
#Funcion Recursiva

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["huellas"])"""

#Esto es para hacer que se ejecute p[or linea de comandos con (-- huellas #detortugas)

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()
```

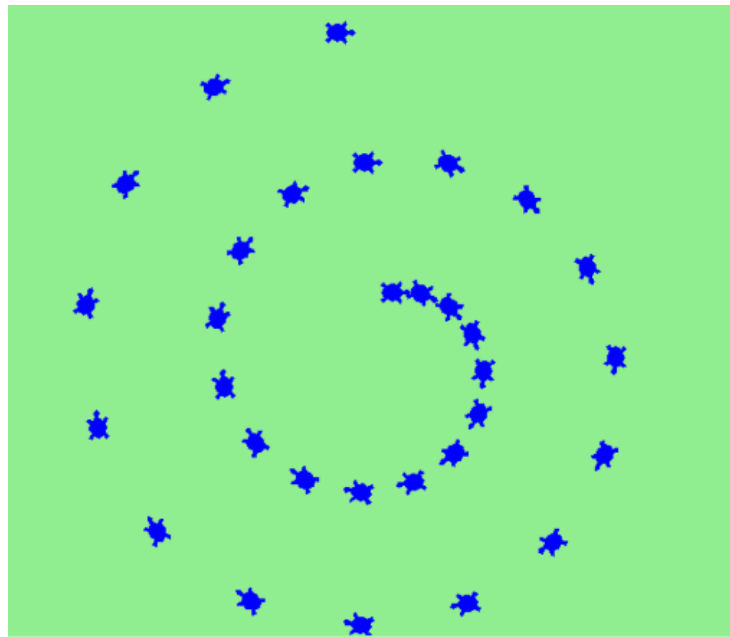
```
import turtle
#funcion iterativa

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()
```

E3.py 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.



Conclusión.

El uso de la recursividad como una herramienta, es una forma muy eficiente que nos ayuda a diseñar y resolver diferentes problemas de una forma sencilla, aunque a veces se nos pueda complicar si no tenemos una visión y algunos problemas sean difíciles de resolver de forma iterativa, ya que esta forma suele ocupar mas elementos para su programación.