



Carátula para entrega de prácticas

Facultad de Ingeniería

Laboratorio de docencia

Laboratorios de computación salas A y B

Ing. Adrián Ulises Mercado Martínez

Profesor:

Estructura de Datos y Algoritmos I

Asignatura:

13

Grupo:

Practica 12. Recursividad

No de Práctica(s):

Forbes Guzmán Logan Aubrey

Integrante(s):

2020-2

Semestre:

CALIFICACIÓN: _____

Introducción:

En esta practica revisamos los conceptos de recursividad, en la recursividad una función se llama a si misma hasta que se logre un objetivo esta usa el concepto de la estrategia divide y vencerás haciendo que los problemas se vuelvan pequeños y fáciles de resolver, en esta practica vamos a aplicarlo en c para insertar un nombre a una lista y en python trabajaremos con una biblioteca en la que podemos jugar con el movimiento de una tortuga

Desarrollo:

En el .h se creo lo usual pero aparte se creo una estructura llamada info donde se guardan los nombres

```
1  #ifndef EJ1_H
2  #define EJ1_H
3
4  typedef struct _node NODE;
5
6  typedef struct _info{
7      char nombre[32];
8      char apellido[64];
9  } INFO;
10
11 typedef struct _node NODE;
12
13 struct _node{
14     INFO info;
15     NODE *next;
16     NODE *prev;
17 };
18
19 typedef struct _list{
20     NODE *tail;
21     NODE *head;
22 } LIST;
23
24 void insertar(INFO info, LIST *l);
25 LIST *crear_lista();
26 void eliminar(LIST *l);
27 void imprimir(LIST *l);
28
29 NODE *crear_nodo();
30 void borrar_nodo(NODE *n);
31 #endif
```

En él .c se desarrollan las funciones es cuando nos damos cuenta de que se esta usando recursividad ya que en las funciones como borrar nodo se llama dentro de esta a borrar nodo de nuevo

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  #include <string.h>
4  #include "ej1.h"
5
6  void insertar(INFO info, LIST *l){
7      if(l!=NULL){
8          if(l->head != NULL){
9              l->head = crear_nodo();
10             l->head->info=info;
11             return;
12         }
13     }
```

```

13     NODE *nuevo = crear_nodo();
14     nuevo->info = info;
15     nuevo->next = l->head;
16     l->head->prev = nuevo;
17     l->head = nuevo;
18 }
19 }
20 LIST *crear_lista(){
21     LIST* l = (LIST*) malloc(sizeof(LIST));
22     l->head = NULL;
23     l->tail = NULL;
24     return l;
25 }
26 void eliminar(LIST *l){
27     if(l->head!=NULL){
28         borrar_nodo(l->head);
29     }
30     free(l);
31 }
32
33 NODE * crear_nodo(){
34     NODE *n = (NODE*) malloc(sizeof(NODE));
35     n->next = NULL;
36     n->prev = NULL;
37     strcpy(n->info.nombre, "");
38     strcpy(n->info.apellido, "");
39     return n;
40 }
41 void borrar_nodo(NODE *n){
42     if(n->next!=NULL){
43         borrar_nodo(n->next);
44     }
45     n->prev = NULL;
46     free(n);
47 }
48
49 void imprimir(LIST *l){
50     for(NODE *i = l->head; i!=NULL; i=i->next){
51         printf("%s, %s\n", i->info.nombre, i->info.apellido);
52     }
53 }

```

```

1  #include <stdio.h>
2  #include <string.h>
3  #include "ej1.h"
4
5  int main(){
6      LIST *lista;
7      INFO info;
8      strcpy(info.nombre, "nombre1");
9      strcpy(info.apellido, "apellido11 apellido 12");
10
11      lista = crear_lista();
12      insertar(info, lista);
13      imprimir(lista);
14      eliminar(lista);
15      return 0;
16 }

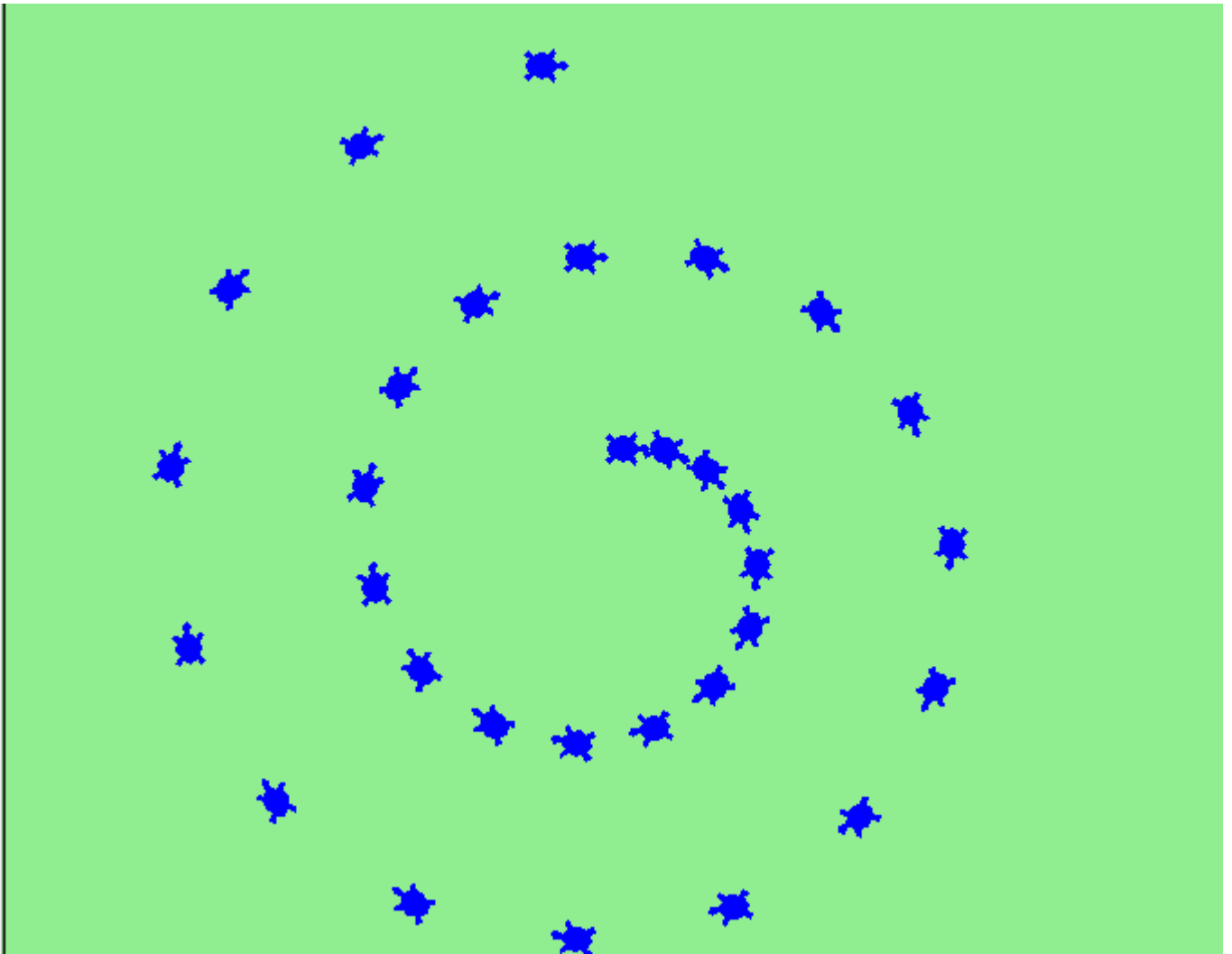
```

En este programa se usa la biblioteca turtle pero no es una función recursiva

```
1  import turtle
2
3  wn = turtle.Screen()
4  wn.bgcolor("lightgreen")
5  wn.title("Tortuga")
6  tess = turtle.Turtle()
7  tess.shape("turtle")
8  tess.color("blue")
9
10 tess.penup()
11 size = 20
12 for i in range(30):
13     tess.stamp()
14     size = size + 3
15     tess.forward(size)
16     tess.right(24)
17
18 wn.mainloop()
```

En este programa se usa la biblioteca turtle pero ahora si es una función recursiva, ya que se llama a si mismo.

```
1  import turtle
2
3  def recorrido_recursivo(tortuga, espacio, huellas):
4      if huellas > 0:
5          tortuga.stamp()
6          espacio = espacio+3
7          tortuga.forward(espacio)
8          tortuga.right(24)
9          recorrido_recursivo(tortuga, espacio, huellas-1)
10
11
12 wn = turtle.Screen()
13 wn.bgcolor("lightgreen")
14 wn.title("Tortuga")
15 tess = turtle.Turtle()
16 tess.shape("turtle")
17 tess.color("blue")
18
19 tess.penup()
20 recorrido_recursivo(tess, 20, 30)
21
22
23 wn.mainloop()
```



Conclusiones:

Forbes Guzmán Logan Aubrey

Por lo que entendí una función recursiva tiene la función de facilitar mucho los programas ya que ahorran varias líneas del programa pero creo que no se deben de utilizar todo el tiempo creo que es algo que también debemos de analizar