|  |  |
| --- | --- |
|  | **Caratula para entrega de Prácticas** |
| Facultad de Ingeniería | Laboratorio de docencia |

Laboratorios de computación salas A y B

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Profesor:\_\_\_\_Ing. Marco Antonio Martinez Quintana\_\_\_

Asignatura:\_Estructura de Datos y Algoritmos I \_

Grupo:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_17\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No. de Práctica(s):\_\_\_\_\_\_\_\_\_\_\_3\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Integrante(s):Francisco Moises Barrera Guardia

No. de Equipo de  
 cómputo empleado:\_\_\_\_\_\_\_\_\_\_13\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

No. Lista o Brigada:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Semestre:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_2°\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Fecha de entrega:\_\_\_Febrero del 2020\_\_\_\_\_\_\_\_\_

Observaciones:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

CALIFICACIÓN:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Objetivo**

Conocer las definiciones, características, procedimientos y ejemplos de las estructuras lineales Pila y Cola, con la finalidad de que comprendas sus estructuras y puedas implementarlas.

**Introducción**

Una estructura de datos consiste en una colección de nodos o registros del mismo tipo que mantienen relaciones entre sí. Un nodo es la unidad mínima de almacenamiento de información en una estructura de datos.

Las estructuras de datos lineales son aquellas en las que los elementos ocupan lugares sucesivos en la estructura y cada uno de ellos tiene un único sucesor y predecesor.

La pila (o stack) es una estructura de datos lineal y dinámica, en la cual el elemento obtenido a través de la operación ELIMINAR está predefinido, debido a que implementa la política Last-In, First-Out (LIFO), esto es, el último elemento que se agregó es el primer que se elimina.

Las operaciones que se pueden realizar sobre una pila son INSERTAR (que es llamada PUSH) y ELIMINAR (que es llamada POP). Debido a la política LIFO que implementa esta estructura, el orden en el que los elementos son extraídos de la pila (POP) es inverso al orden en el que los elementos fueron insertados en la pila (PUSH). Además, el único elemento accesible de la pila es el que está hasta arriba y que se conoce como tope de la pila.

La cola (o queue) es una estructura de datos lineal, en la cual el elemento obtenido a través de la operación ELIMINAR está predefinido y es el que se encuentra al inicio de la estructura.

La cola implementa la política First-In, First-Out (FIFO), esto es, el primer elemento que se agregó es el primero que se elimina.

La cola es una estructura de datos de tamaño fijo y cuyas operaciones se realizan por ambos extremos; permite INSERTAR elementos al final de la estructura y permite ELIMINAR elementos por el inicio de la misma. La operación de INSERTAR también se le llama ENCOLAR y la operación de ELIMINAR también se le llama DESENCOLAR.

**Desarrollo**

En esta práctica se elaboró un programa que hiciera las funciones de una pila con el código previamente elaborado de nuestra tienda que se elaboró en la práctica 3. Entonces en este programa el usuario ingresaba los datos para comprar un coche y esa estructura se guardaba, el usuario tenía la opción de mostrar la pila, ingresar un dato nuevo eliminar un dato y finalizar el programa.

**Resultados:**

**Código:**

**#include <stdio.h>**

**#include <stdlib.h>**

**struct Lamborghini**

**{**

**char Nombre[20];**

**int Precio;**

**int Inventario;**

**char SKU[30];**

**char Color[15];**

**struct Lamborghini \*ptrSiguiente;**

**};**

**void push(struct Lamborghini \*\*ptrS);**

**void pop(struct Lamborghini \*\*ptrS);**

**void mostrarPila (struct Lamborghini \*ptrS);**

**int main(){**

**struct Lamborghini \*ptrHead;**

**int opcion = 1;**

**ptrHead = NULL;**

**while (opcion != 0) {**

**printf("Selecciona una opcion\n");**

**printf("1.- Ingresar coche\n2.- Eliminar coche\n3.- Mostrar los coches\n0.- Salir\n");**

**scanf("%d", &opcion);**

**fflush(stdin);**

**switch (opcion) {**

**case 1:**

**push(&ptrHead);**

**break;**

**case 2:**

**pop(&ptrHead);**

**break;**

**case 3:**

**mostrarPila(ptrHead);**

**break;**

**default:**

**break;**

**}**

**}**

**return 0;**

**}**

**void push (struct Lamborghini \*\*ptrS)**

**{**

**struct Lamborghini \*ptrNuevo;**

**ptrNuevo = (struct Lamborghini \*)malloc(sizeof(struct Lamborghini));**

**if(ptrNuevo != NULL)**

**{**

**printf("\t\tCompra de autos\n");**

**printf("Ingrese el Nombre del auto que desee comprar:\n");**

**scanf("%s", ptrNuevo->Nombre);**

**fflush(stdin);**

**printf("Ingrese el precio del auto que desee comprar:\n");**

**scanf("%i", &ptrNuevo->Precio);**

**fflush(stdin);**

**printf("Ingrese el inventario del auto que desee comprar:\n");**

**scanf("%i", &ptrNuevo->Inventario);**

**fflush(stdin);**

**printf("Ingrese el SKU del auto que desee comprar:\n");**

**scanf("%s", ptrNuevo->SKU);**

**fflush(stdin);**

**printf("Ingrese el color del auto que desee comprar:\n");**

**scanf("%s", ptrNuevo->Color);**

**fflush(stdin);**

**ptrNuevo->ptrSiguiente = \*ptrS;**

**\*ptrS = ptrNuevo;**

**}else{**

**printf("no se creo el nuevo dato\n");**

**}**

**}**

**void pop(struct Lamborghini \*\*ptrS){**

**struct Lamborghini \*prtTemp;**

**prtTemp = \*ptrS;**

**\*ptrS = (\*ptrS)->ptrSiguiente;**

**free(prtTemp);**

**}**

**void mostrarPila (struct Lamborghini \*ptrS){**

**struct Lamborghini \*ptrActual;**

**if (ptrS == NULL){**

**printf ("pila Vacia\n");**

**return;**

**}**

**ptrActual = ptrS;**

**printf("La Pila Actual es:\n");**

**while (ptrActual != NULL){**

**printf("%s -> ", ptrActual->Nombre);**

**printf("%d -> ", ptrActual->Precio);**

**printf("%d -> ", ptrActual->Inventario);**

**printf("%s -> ", ptrActual->SKU);**

**printf("%s -> ", ptrActual->Color);**

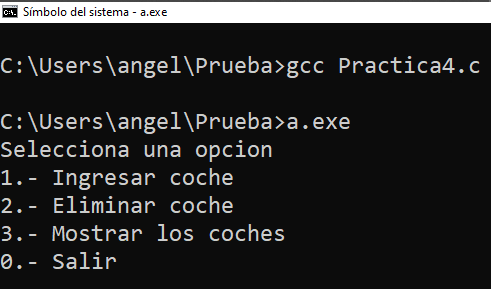
**ptrActual = ptrActual->ptrSiguiente;**

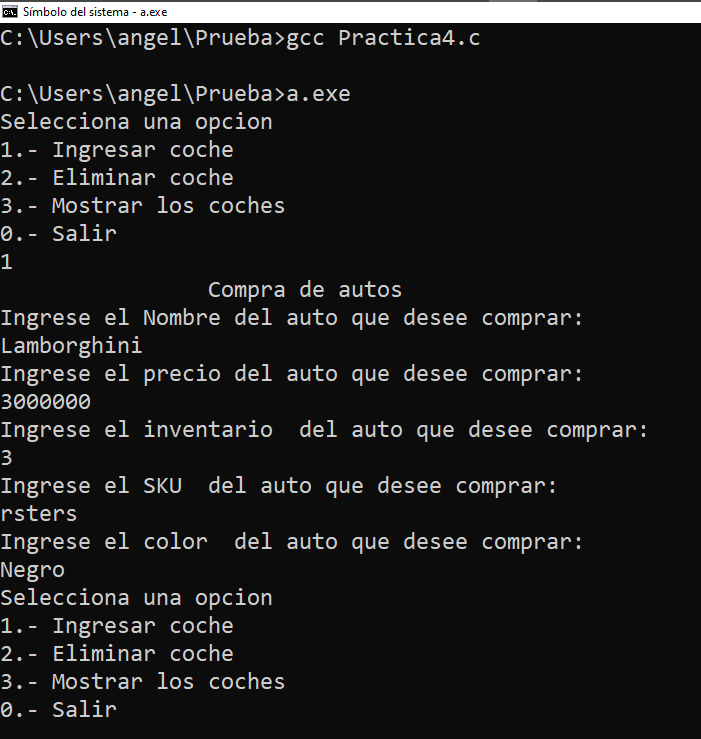
**}**

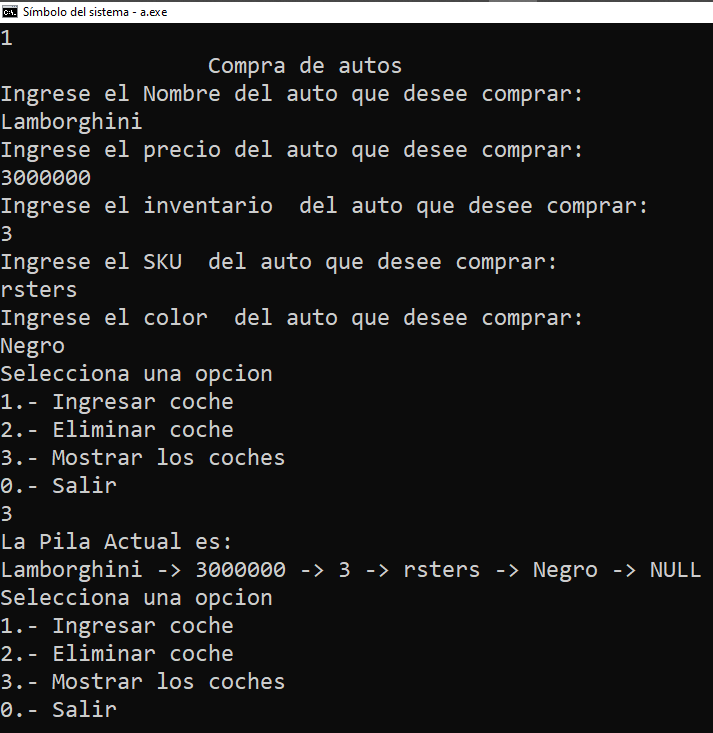
**printf("NULL\n");**

**}**

**Ejemplo de Terminal :**







**Conclusiones:**

En esta práctica se aprecia el funcionamiento de la Pila aplicada a estructuras y que no solo es aplicable a variables de tipo carácter o enteros, además, de la funcionalidad de las pilas dentro de una lista o una memoria dinámica en la cual se buscan apilar los valores y después poder quitarlos uno por uno.