



Universidad Autónoma de Baja California
Facultad de Ciencias Químicas e Ingeniería
Algoritmos y Estructuras de Datos
Ciclo 2020-1

Práctica 1
“Estructuras LIFO y FIFO Estáticas”

Profesor(a): Thelma Violeta Ocegueda Miramontes

Grupo: 552

Alumno:

- Rodríguez Muñoz José Luis

Matrícula:

1260368

Tijuana, Baja California, a 07 de febrero de 2020

CÓDIGO EN C

```
#include <stdio.h> // Librería principal
#include <conio.h> // Librería para el uso de getch
#define MAX 5 // Valor máximo para la pila y cola

// Estructura para manejar la pila estática
typedef struct
{
    int tope; // Tope de la pila
    int dato[MAX]; // Capacidad de la pila
} Pila;

Pila pila; // Atributo de la estructura Pila

// Estructura para manejar la cola estática
typedef struct
{
    int datos[MAX]; // Capacidad de la cola
    int inicio, final; // Inicio y final de la cola
} Cola;

Cola cola; // Atributo de la estructura cola

// Funciones prototipo
void menuPilas();
void menuColas();
void mostrarPila(Pila * pila);
void mostrarCola(Cola * cola);

int main()
{
    return menu();
}

/* Funciones para el manejo de la Pila */

void push(Pila * pila, int x)
{
    pila -> dato[pila -> tope++] = x; // Inserta un elemento en el tope de la
pila
}

int pop(Pila * pila)
{
    return (pila -> dato[--pila -> tope]); // Regresa el último elemento
insertado en la pila
}

int vacia(Pila * pila)
{
    if(pila -> tope == 0)
        return 1;
    else
        return 0; // Regresa verdadero si la pila está vacía
}
```

```

int llena(Pila * pila)
{
    if(pila -> tope == MAX)
        return 1;
    else
        return 0; // Regresa verdadero si la pila está llena
}

// Función para mostrar la pila
void mostrarPila(Pila * pila)
{
    int i;

    for(i = pila -> tope - 1; i > - 1; i--)
    {
        printf("\n [ %d ]", pila->dato[i]); // Se imprime el arreglo
    }

    if(!pila -> tope)
    {
        printf("\n Pila vacía ...", 161); // Si es diferente de tope está
vacía
    }
}

/* Funciones para el manejo de la Cola */

// Función para crear la cola
void crearCola(Cola * cola)
{
    cola -> inicio = 0; // Se crea cola vacía
    cola -> final = 0;
}

int llenaCola(Cola cola)
{
    if(cola.final == MAX) // Regresa verdadero si la cola está llena
        return 1;

    else
        return 0;
}

int vaciaCola(Cola cola)
{
    if(cola.final == 0) // Regresa verdadero si la cola está vacía
        return 1;

    else
        return 0;
}

int remover(Cola * cola)
{
    int x, i;
    x = cola -> datos[0];

```

```

    for(i = 0; i < cola -> final - 1; i++)
    {
        cola -> datos[i] = cola -> datos[i + 1]; // Este bloque de código
remueve el último elemento insertado en la cola
    }
    cola -> final--;
    return x;
}

void insertar(Cola * cola, int dato)
{
    cola -> datos[cola -> final] = dato;
    cola -> final++; // Inserta un elemento al inicio
de la cola
}

// Función para mostrar la cola
void mostrarCola(Cola * cola)
{
    int i;

    printf("\n"); // Estética
    for(i = 0; i < cola -> final; i++)
    {
        printf(" [ %d ] ", cola->datos[i]); // Se imprime el arreglo
    }

    if(!cola -> final)
    {
        printf("\n Cola vac%ca ...", 161); // Si es diferente de final está
vacía
    }
}

// Función para manejar el menú principal
int menu()
{
    char opcion;

    do
    {
        system("cls"); // Limpia pantalla
        printf("\n\t Estructuras LIFO y FIFO Est%cticas", 160);
        printf("\n\n 1) Manejo de Pilas");
        printf("\n 2) Manejo de Colas");
        printf("\n 3) Salir");
        printf("\n\n\t Elige una opci%cn: ", 162);
        opcion = getchar(); // Captura de opción

        switch(opcion)
        {
            case '1':
                system("cls"); // Limpia pantalla
                menuPilas(); // Abre el menú de pilas
                break; // Rompe iteración

```

```

        case '2':
            system("cls"); // Limpia pantalla
            menuColas(); // Abre menú de colas
            break; // Rompe iteración
    }
} while (opcion != '3' && opcion != EOF);
{
    system("cls");
    printf("\n Cierre exitoso ..."); // Termina el programa
}

// Manejo del submenú de Pila
void menuPilas()
{
    pila.tope = 0; // Se crea una pila vacía
    char opcion;
    int numero;

    do
    {
        system("cls"); // Limpia pantalla
        printf("\n\t Pila Estática", 160);
        printf("\n\n 1) Insertar un elemento (Push)");
        printf("\n 2) Remover un elemento (Pop)");
        printf("\n 3) Tamaño de la pila", 164);
        printf("\n 4) Mostrar pila");
        printf("\n 5) Regresar");
        printf("\n\n\t Introduce una opción: ", 162);
        opcion = getchar(); // Captura de opción

        switch(opcion)
        {
            case '1':
                system("cls"); // Limpia pantalla

                // Si no está llena, puede ingresar dato
                if(!llena(&pila))
                {
                    printf("\n Dato que desea ingresar: "); // Se pide el dato
                    scanf("%d", &numero); // Se lee el dato
                    push(&pila, numero); // Inserta elemento
                }
                // En caso contrario, la pila está llena
                else
                {
                    printf("\n Pila llena ..."); // Monitoreo
                    getch(); // Hace la pausa hasta presionar enter
                }
                break; // Rompe iteración

            case '2':
                system("cls"); // Limpia pantalla

                // Si no está vacía, puede hacer pop
                if(!vacía(&pila))
                {

```

```

        printf("\n Dato extra%cdo: %d", 161, pop(&pila)); // Remueve
elemento
        getch(); // Hace la pausa hasta presionar enter
    }
    // En caso contrario, la pila está vacía
    else
    {
        printf("\n Pila vac%ca ...", 161); // Monitoreo
        getch(); // Hace la pausa hasta presionar enter
    }
    break; // Rompe iteración

    case '3':
        system("cls"); // Limpia pantalla
        printf("\n Hay %d datos", pila.tope); // Regresa el tope
        getch(); // Hace la pausa hasta presionar enter
        break; // Rompe iteración

    case '4':
        system("cls");
        mostrarPila(&pila); // Muestra la pila
        getch(); // Hace la pausa hasta presionar enter
        break; // Rompe iteración
    }
} while (opcion != '5' && opcion != EOF);
{
    return; // Regresa al menú principal
}
}

// Manejo del submenú de Cola
void menuColas()
{
    char opcion;
    int numero;
    crearCola(&cola); // Se crea la cola

    do
    {
        system("cls"); // Limpia pantalla
        printf("\n\t Cola Est%ctica", 160);
        printf("\n\n 1) Insertar un elemento");
        printf("\n 2) Remover un elemento");
        printf("\n 3) Tama%co de la cola", 164);
        printf("\n 4) Mostrar cola");
        printf("\n 5) Regresar");
        printf("\n\n\t Introduce una opci%cn: ", 162);
        opcion = getchar(); // Captura de opción

        switch(opcion)
        {
            case '1':
                system("cls"); // Limpia pantalla

                // Si está vacía, se puede insertar elemento
                if(!llenaCola(cola))
                {

```

```

        printf("\n Dato que desea ingresar: "); // Se pide dato a
insertar
        scanf("%d", &numero); // Se lee el dato
        insertar(&cola, numero); // Inserta elemento
    }
    // En caso contrario, está llena
    else
    {
        printf("\n Cola llena ..."); // Monitoreo
        getch(); // Hace la pausa hasta presionar enter
    }
    break;

    case '2':
        system("cls"); // Limpia pantalla

        // Si hay datos, se puede extraer
        if(!vacíaCola(cola))
        {
            printf("\n Dato extraído: %d", 161, remover(&cola));
            getch();
        }
        // En caso contrario, no hay ningún dato
        else
        {
            printf("\n No hay datos ..."); // Monitoreo
            getch(); // Hace la pausa hasta presionar enter
        }
        break; // Rompe iteración

    case '3':
        system("cls"); // Limpia pantalla
        printf("\n Hay %d datos", cola.final); // Regresa el final de la
cola
        getch(); // Hace la pausa hasta presionar enter
        break; // Rompe iteración

    case '4':
        system("cls"); // Limpia pantalla
        mostrarCola(&cola); // Muestra la cola
        getch(); // Hace la pausa hasta presionar enter
        break; // Rompe iteración
    }
} while (opcion != '5' && opcion != EOF);
{
    return; // Regresa al menú principal
}
}

```

CONCLUSIÓN

Este ejercicio o bien, práctica, me sirvió para conocer el funcionamiento y la implementación de las estructuras LIFO y FIFO estáticas en un programa en lenguaje C. Una de las observaciones es que son estructuras un poco similares, aunque se diferencian en que, en las pilas, el último elemento insertado es el primero que saldrá al momento de remover un dato de la pila, una representación de la vida cotidiana podría ser cuando lavas platos. Y en el caso de las colas, el primer elemento que entra, será el primero en salir, por ejemplo, cuando hacemos fila para entrar al cine.