Universidad Nacional Autónoma de Nicaragua, León

Facultad de Ciencias y Tecnología

Departamento de Computación

Componente: Algoritmo y Estructura de Datos

Unidad IV: ESTRUCTURAS DINÁMICAS DE DATOS

TEMA: LISTAS DOBLEMENTE ENLAZADAS

Elaborado por:

- DAVID MARADIAGA GUTIÉRREZ.

Fecha:

- 09 de Noviembre, 2020

"A la libertad por la Universidad"

TABLA DE CONTENIDOS

Contenido

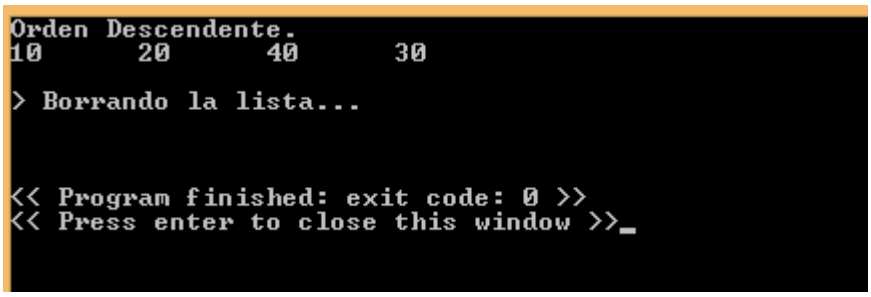
Ejercicio 1.

Ejercicio que pone a prueba las operaciones básicas de una lista doblemente enlazada.

```
#include<stdio.h>
#include<stdlib.h>

struct datos
{
	int valor;
	struct datos *sig, *ant;
};

typedef struct datos *pNodo, *Lista;

pNodo NuevoNodo()
{
	pNodo q = (pNodo)malloc(sizeof(struct datos));
	if(!q) exit(-1);

	return q;
}

void InsertarLista(Lista *cab, int v)
{
	pNodo p = *cab;
	pNodo q = NuevoNodo();

	q->valor = v;

	if(!p)
	{
		q->sig = q->ant = NULL;
	}
	else
	{
		while(p->ant) p = p->ant;

		q->sig = p;
		q->ant = p->ant;
		p->ant = q;
	}

	*cab = q;
	return;
}

void MostrarLista(Lista cab, int orden)
{
	if(orden == 0)    //mostrar de inicio a fin...
	{
```

```
		puts("Orden Ascendente.");
		while(cab->ant) cab = cab->ant;

		while(cab)
		{
			printf("%d\t", cab->valor);
			cab = cab->sig;
		}
	}
	else	//mostrar de fin a inicio...
	{
		puts("Orden Descendente.");
		while(cab->sig) cab = cab->sig;

		while(cab)
		{
			printf("%d\t", cab->valor);
			cab = cab->ant;
		}
	}
}

void BorrarLista(Lista *cab)
{
	pNodo q = *cab;

	while(q)
	{
		*cab = q->sig;
		free(q);
		q = *cab;
	}
	printf("\n\n> Borrando la lista...\n\n");
}

int main()
{
	Lista c = NULL;

	InsertarLista(&c, 10);
	InsertarLista(&c, 20);
	InsertarLista(&c, 40);
	InsertarLista(&c, 30);

	MostrarLista(c, 1);	//mostr

	BorrarLista(&c);

	return 0;
}
```

Salida de ejecución:



Actividad: Modificar este ejemplo para trabajarlo con un menú de opciones y añada la operación de borrar un nodo cualquiera.

Ejercicio 2.

Una compañía distribuye productos a distintos comercios de la cuidad. La información suministrada de los productos es la siguiente:

- Clave.
- Descripción.
- Existencia (valor mínimo: 5).
- Precio Unitario.

Elabore un programa que permita implementar las operaciones básicas de una lista doblemente enlazada.

```
#include<stdio.h>
#include<stdlib.h>

//declaracion del tipo producto
struct producto
{
  int clave;
  char *descripcion;
  int existencia;
  float precio;
  struct datos *sig, *ant;
};

typedef struct producto *pNodo, *Lista;

//prototipos de funciones
```

```
pNodo NuevoNodo();
void Error();
void InsertarLista(Lista *cab, int c, char *desc, int exi, float p);
void MostrarLista(Lista cab, int orden);
void BorrarLista(Lista *cab);

//COMPLETAR EL CODIGO…
```

Ejercicio 3.

Modificar el ejercicio anterior, añadiendo la siguiente funcionalidad:

- Ventas de un producto: Se debe actualizar los que correspondan, y verifique que la nueva existencia no esté por debajo del mínimo (Datos: clave, cantidad vendida).
- Reabastecimiento de un producto: Se deben actualizar los campos que correspondan. (Datos: Clave, Cantidad comprada).
- Información sobre el producto: Se deben proporcionar todos los datos relacionados a un producto. (Dato: Clave).