/\*\*Este Proyecto es un Repaso del Tema de Listas (Linked Lists)\*\*/

#include <stdio.h>

#include <stdlib.h>

typedef struct element\_of\_a\_list{

    int iData;

    struct element\_of\_a\_list \*next;

}node;

int main()

{

    node \*startOfList = createLinkedList();

    printLinkedList(startOfList);

    printf("\nLa lista tiene %i elementos. \n", numberOfElementsInList(startOfList));

    printIfNumberExists(startOfList);

    printf("\n \n");

    return 0;

}

/\*\*Este Procedimiento se Encarga de Crear e Imprimir un Lista Vinculada\*\*/

#include<stdio.h>

#include<stdlib.h>

typedef struct element\_of\_a\_list{

    int iData;

    struct element\_of\_a\_list \*next;

}node;

void printLinkedList(node \*head)

{

    printf("\nImpresion de los Elementos de la Lista. \n");

    node \*current\_node = head;

    while (current\_node != NULL)

    {

        printf("\tElemento: %i\n", current\_node->iData);

        current\_node = current\_node->next;

    }

}

node \*createLinkedList()

{

    printf("Ingreso de Elementos de la Lista. \n");

    node \*head;

    node \*current\_node;

    int iNumber;

    printf("\tIngresa un numero o ingresa '-1' para terminar el proceso: ");

    scanf("%i", &iNumber);

    if (iNumber == -1)

        return NULL;

    head = malloc(sizeof(node));

    current\_node = head;

    current\_node->iData = iNumber;

    printf("\tIngresa un numero o ingresa '-1' para terminar el proceso: ");

    scanf("%i", &iNumber);

    while (iNumber != -1)

    {

        current\_node->next = malloc(sizeof(node));

        current\_node = current\_node->next;

        current\_node->iData = iNumber;

        printf("\tIngresa un numero o ingresa '-1' para terminar el proceso: ");

        scanf("%i", &iNumber);

    }

    current\_node->next = NULL;

    return head;

}

void callProceduresOfLists()

{

    printLinkedList(createLinkedList());

}

/\*\*Este Procedimiento se Encarga de Calcular el Numero de Elementos en una Lista Dinamica\*\*/

#include<stdio.h>

#include<stdlib.h>

typedef struct element\_of\_a\_list{

    int iData;

    struct element\_of\_a\_list \*next;

}node;

int numberOfElementsInList(node \*head)

{

    int iCounter = 0;

    node \*current\_node = head;

    while (current\_node != NULL)

    {

        iCounter++;

        current\_node = current\_node->next;

    }

    return iCounter;

}

/\*\*Este Procedimiento Determina si un Numero Est� Presente en la Lista Din�mica\*\*/

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

typedef struct element\_of\_a\_list{

    int iData;

    struct element\_of\_a\_list \*next;

}node;

bool numberExistsInListQuestion(node \*head, int iNumber)

{

    node \*current\_node = head;

    while (current\_node != NULL)

    {

        if (iNumber == current\_node->iData)

            return true;

        current\_node = current\_node->next;

    }

    return false;

}

void printIfNumberExists(node \*head)

{

    int iNumber;

    printf("\nIngresa un numero: ");

    scanf("%i", &iNumber);

    if (numberExistsInListQuestion(head, iNumber))

    {

        printf("El numero, %i, existe en la lista. \n", iNumber);

        printf("Total apariciones: %i. \n", numberOfAppearancesOfNumberInList(head, iNumber));

    }

    else

        printf("El numero, %i, no existe en la lista. \n", iNumber);

}

/\*\*Este Procedimiento Determina el Numero de Veces que un Numero se Repite Dentro de una Lista\*\*/

#include<stdio.h>

#include<stdlib.h>

#include<stdbool.h>

typedef struct element\_of\_a\_list{

    int iData;

    struct element\_of\_a\_list \*next;

}node;

int numberOfAppearancesOfNumberInList(node \*head, int iNumber)

{

    int iCounter = 0;

    node \*current\_node = head;

    while (current\_node != NULL)

    {

        if (iNumber == current\_node->iData)

            iCounter++;

        current\_node = current\_node->next;

    }

    return iCounter;

}

/\*\*Este Procedimiento nos Ense�a 1 Forma de Usar el Tipo de Dato BOOL\*\*/

#include<stdio.h>

#include<stdlib.h>

/// 1era Forma

#include<stdbool.h>