

```

/**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
Dinamica**/

#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 Enseña 1 Forma de Usar el Tipo de Dato BOOL**/

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

/// 1era Forma
#include<stdbool.h>

```

