**TITLE 42**

Write a C program to search an element in a singly-linked list

**OBJECTIVE:**

By the end of this problem we will be able to search an element in a singly-linked list

**PROBLEM STATEMENT:**

In this problem we search an element in a singly-linked list. Input from the user:

Enter number:

Once the input is entered and stored, the output is printed.

**ALGORITHM:**

START

Define variables: num, \*next

INPUT: Read from the keyboard

COMPUTATION: Computing the search of an element in a singly-linked list

DISPLAY: Displaying the element

STOP

**PROGRAM:**

#include <stdio.h>

#include <stdlib.h>

struct node

{

int num;

struct node \*next;

};

void create(struct node \*\*);

int search(struct node \*, int);

void release(struct node \*\*);

void display(struct node \*);

int main()

{

struct node \*p = NULL;

int key, result;

printf("Enter data into the list\n");

create(&p);

printf("Displaying the nodes in the list:\n");

display(p);

printf("Enter key to search in the list: ");

scanf("%d", &key);

result = search(p, key);

if (result)

{

printf("%d found in the list.\n", key);

}

else

{

printf("%d not found in the list.\n", key);

}

release(&p);

return 0;

}

int search(struct node \*head, int key)

{

while (head != NULL)

{

if (head->num == key)

{

return 1;

}

head = head->next;

}

return 0;

}

void create(struct node \*\*head)

{

int c, ch;

struct node \*temp, \*rear;

do

{

printf("Enter number: ");

scanf("%d", &c);

temp = (struct node \*)malloc(sizeof(struct node));

temp->num = c;

temp->next = NULL;

if (\*head == NULL)

{

\*head = temp;

}

else

{

rear->next = temp;

}

rear = temp;

printf("Do you wish to continue [1/0]: ");

scanf("%d", &ch);

} while (ch != 0);

printf("\n");

}

void display(struct node \*p)

{

while (p != NULL)

{

printf("%d\t", p->num);

p = p->next;

}

printf("\n");

}

void release(struct node \*\*head)

{

struct node \*temp = \*head;

\*head = (\*head)->next;

while ((\*head) != NULL)

{

free(temp);

temp = \*head;

(\*head) = (\*head)->next;

}

}

**CONCLUSION:**

The simulation of the C program helped me understand how we can search an element in a singly-linked list.

**OUTPUT:**

Enter data into the list

Enter number: 4

Do you wish to continue [1/0]: 1

Enter number: 5

Do you wish to continue [1/0]: 1

Enter number: 6

Do you wish to continue [1/0]: 0

Displaying the nodes in the list:

4 5 6

Enter key to search in the list: 5

5 found in the list.