EXPERIMENT – 1:

**1. Write an algorithm and program to implement Linear Search. i) Using arrays Ii) Using linked list**

**2. Write an algorithm and program in to implement Binary Search. i) Using arrays Ii) Using linked list**

LINEAR SEARCH USING LINKED LIST:

#include <stdio.h>

#include <stdlib.h>

struct node

{

int data;

struct node \*next;

} \* head;

void createList(int n);

void displayList();

int search(int key);

int main()

{

int n, keyToSearch, index;

printf("Enter number of node to create: ");

scanf("%d", &n);

createList(n);

printf("\nData in list: \n");

displayList();

printf("\nEnter element to search: ");

scanf("%d", &keyToSearch);

index = search(keyToSearch);

if (index >= 0)

printf("%d found in the list at position %d\n", keyToSearch, index + 1);

else

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

return 0;

}

void createList(int n)

{

struct node \*newNode, \*temp;

int data, i;

head = malloc(sizeof(struct node));

if (head == NULL)

{

printf("Unable to allocate memory. Exiting from app.");

exit(0);

}

printf("Enter data of node 1: ");

scanf("%d", &data);

head->data = data;

head->next = NULL;

temp = head;

for (i = 2; i <= n; i++)

{

newNode = malloc(sizeof(struct node));

/\* If memory is not allocated for newNode \*/

if (newNode == NULL)

{

printf("Unable to allocate memory. Exiting from app.");

exit(0);

}

printf("Enter data of node %d: ", i);

scanf("%d", &data);

newNode->data = data;

newNode->next = NULL;

temp->next = newNode;

temp = temp->next;

}

}

void displayList()

{

struct node \*temp;

if (head == NULL)

{

printf("List is empty.\n");

return;

}

temp = head;

while (temp != NULL)

{

printf("%d, ", temp->data);

temp = temp->next;

}

printf("\n");

}

int search(int key)

{

int index;

struct node \*curNode;

index = 0;

curNode = head;

while (curNode != NULL && curNode->data != key)

{

index++;

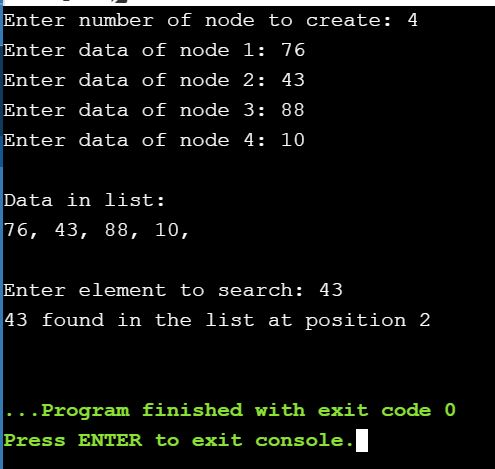
curNode = curNode->next;

}

return (curNode != NULL) ? index : -1;

}

OUTPUT:



BINARY SEARCH USING LINKED LIST:

#include<stdio.h>

#include<stdlib.h>

struct Node

{

int data;

struct Node\* next;

};

Node \*newNode(int x)

{

struct Node\* temp = new Node;

temp->data = x;

temp->next = NULL;

return temp;

}

struct Node\* middle(Node\* start, Node\* last)

{

if (start == NULL)

return NULL;

struct Node\* slow = start;

struct Node\* fast = start -> next;

while (fast != last)

{

fast = fast -> next;

if (fast != last)

{

slow = slow -> next;

fast = fast -> next;

}

}

return slow;

}

struct Node\* binarySearch(Node \*head, int value)

{

struct Node\* start = head;

struct Node\* last = NULL;

do

{

Node\* mid = middle(start, last);

if (mid == NULL)

return NULL;

if (mid -> data == value)

return mid;

else if (mid -> data < value)

start = mid -> next;

else

last = mid;

} while (last == NULL ||

last != start);

return NULL;

}

int main()

{

Node \*head = newNode(1);

head->next = newNode(4);

head->next->next = newNode(7);

head->next->next->next = newNode(8);

head->next->next->next->next = newNode(9);

head->next->next->next->next->next = newNode(10);

int value = 7;

if (binarySearch(head, value) == NULL)

printf("Value not present");

else

printf("Present");

return 0;

}

MENU DRIVEN FOR SEARCHING IN ARRAY:

#include<stdio.h>

#include<conio.h>

void main()

{

clrscr();

int arr[10],n,i,c,n1,first,last,middle;

printf("\nEnter size of array:");

scanf("%d",&n);

printf("\nEnter the numbers of the array:");

for(i=0;i<n;i++)

{

scanf("%d",&arr[i]);

}

printf("Select one of the options:\n");

printf("\n1.LINEAR SEARCH\n2.BINARY SEARCH\n");

scanf("%d",&c);

switch(c)

{

case 1:

printf("LINEAR SEARCH\n");

printf("\nEnter number you want to search in the array:");

scanf("%d",&n1);

for(i=0;i<n;i++)

{

if(n1==arr[i])

{

printf("\nElement found at index %d.",i);

break;

}

}

break;

case 2:

printf("BINARY SEARCH\n");

printf("\nEnter number you want to search in the array:");

scanf("%d",&n1);

first=0;

last=n-1;

middle=(first + last)/2;

while(first<=last)

{

if(arr[middle]<n1)

{

first=middle+1;

}

else if(arr[middle]==n1)

{

printf("%d found at location %d.\n",n1,middle);

break;

}

else

{

last=middle-1;

}

middle=(first+last)/2;

}

if (first>last)

{

printf("Not found! %d isn't present in the list.\n",n1);

}

break;

}

getch();

}

OUTPUT:

