

# Nithin S

## 221IT085

Q1) Implement Binary Search using linked lists and Array.

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

struct Node
{
    int data;
    struct Node *next;
};

void insertLast(struct Node **head,int x)
{
    struct Node* temp=(struct Node*)malloc(sizeof(struct Node));
    temp->data=x;
    temp->next=NULL;
    if(*head==NULL) *head=temp;
    else
    {
        struct Node *p=*head;
        while(p->next)
        {
            p=p->next;
        }
        p->next=temp;
    }
}

struct Node* findMiddle(struct Node*head,struct Node*end)
{
    if(head==NULL) return NULL;
    struct Node *fast=head->next;
```

```

struct Node *slow=head;
while(fast!=end)
{
    fast=fast->next;
    if(fast!=end)
    {
        fast=fast->next;
        slow=slow->next;
    }
}
return slow;
}

```

```

void display(struct Node *p)
{
    while(p)
    {
        printf("%d ",p->data);
        p=p->next;
    }
}

```

```

struct Node* binarySearch(struct Node*head,int key)
{
    struct Node*first=head;
    struct Node*last=NULL;
    do
    {
        struct Node* mid=findMiddle(first,last);
        if(mid==NULL) return NULL;
        if(mid->data==key) return mid;
        else if(mid->data>key) last=mid;
        else first=mid->next;
    } while (last==NULL||last!=first);
    return NULL;
}

```

```

bool binarySearchArray(int arr[],int key,int n)
{
    int start=0;
    int last=n-1;
    while(start<=last)

```

```

{
    int mid=(start+last)/2;
    if(arr[mid]==key) return true;
    else if(arr[mid]>key) last=mid-1;
    else start=mid+1;
}
return false;
}

```

```

int main()
{
    int n=0;
    int key=0;
    char choice;
    printf("Do you want to search in an Array(A) or linkedList(L).Enter 'L' or 'A':
");
    scanf("%c",&choice);
    if(choice=='L')
    {
        struct Node *ll1=NULL;
        printf("Enter the No. of elements in the linked list: ");
        scanf("%d",&n);
        printf("Enter the elements (Integers) seperated by spaces: ");
        while(n>0)
        {
            int num=0;
            scanf("%d",&num);
            insertLast(&ll1,num);
            n--;
        }
        printf("The entered linkedList: ");
        display(ll1);
        printf("\nEnter an element to search: ");
        scanf("%d",&key);
        if(binarySearch(ll1,key)==NULL) printf("%d is not present in the
linkedList.\n",key);
        else printf("%d is present in the linkedList.\n",key);
        return 0;
    }
}

```

```

else if(choice=='A')
{
    printf("Enter the No. of elements in the array: ");
    scanf("%d",&n);
    int *arr=(int*)malloc(sizeof(int)*n);
    printf("Enter the elements seperated by spaces: ");
    for(int i=0;i<n;i++)
        scanf("%d",&arr[i]);

    printf("The Array contains: ");
    for(int i=0;i<n;i++)
        printf("%d ",arr[i]);
    printf("\nEnter the element to be searched: ");
    scanf("%d",&key);
    if(binarySearchArray(arr,key,n)) printf("%d is present in the linkedList.\n",key);
    else printf("%d is not present in the linkedList.\n",key);
}
else
{
    printf("Invalid Choice");
    return 0;
}
}

```

## OUTPUT

```

student@HP-Elite600G9-10:~/Desktop/Assignment$ gcc 1_BinarySearch.c
student@HP-Elite600G9-10:~/Desktop/Assignment$ ./a.out
Do you want to search in an Array(A) or linkedList(L).Enter 'L' or 'A': L
Enter the No. of elements in the linked list: 5
Enter the elements (Integers) seperated by spaces: 1 5 6 7 8
The entered linkedList: 1 5 6 7 8
Enter an element to search: 5
5 is present in the linkedList.

```

```

student@HP-Elite600G9-10:~/Desktop/Assignment$ gcc 1_BinarySearch.c
student@HP-Elite600G9-10:~/Desktop/Assignment$ ./a.out
Do you want to search in an Array(A) or linkedList(L).Enter 'L' or 'A': A
Enter the No. of elements in the array: 5
Enter the elements seperated by spaces: 1 2 3 4 5
The Array contains: 1 2 3 4 5
Enter the element to be searched: 8
8 is not present in the linkedList.

```

## Q2) Implement Stack using Array

```
#include<stdio.h>
#include<stdlib.h>
#include<stdbool.h>
#define MAX_SIZE 100

int top = -1;
int stack[MAX_SIZE];

void push(int x)
{
    if(top == MAX_SIZE - 1)
    {
        printf("Stack Overflow.\n");
        return;
    }
    else
    {
        top++;
        stack[top] = x;
    }
}

int pop()
{
    if(top == -1)
    {
        printf("Stack Underflow.\n");
        exit(1);
    }
    return stack[top--];
}

bool isEmpty()
{
    return (top == -1);
}

void display()
{

```

```

    for(int i = 0; i <= top; i++)
        printf("%d ", stack[i]);
    printf("\n");
}

int main()
{
    char choice;
    int num;

    while(1)
    {
        printf("Enter:\n1 to Push, 2 to Pop, q to Quit\n");
        scanf(" %c", &choice);

        if(choice == 'q')
        {
            break;
        }
        else if(choice == '1')
        {
            printf("Enter the element to push: ");
            scanf("%d", &num);
            push(num);
            printf("Stack : ");
            display();
        }
        else if(choice == '2')
        {
            pop(num);
            printf("Stack : \n");
            display();
        }
        else
        {
            printf("Invalid choice. Try again.\n");
        }
    }

    return 0;
}

```

# OUTPUT

```
student@HP-Elite600G9-10:~/Desktop/Assignment$ gcc 2_StackUsingArr.c
student@HP-Elite600G9-10:~/Desktop/Assignment$ ./a.out
Enter:
1 to Push, 2 to Pop, q to Quit
1
Enter the element to push: 34
Stack : 34
Enter:
1 to Push, 2 to Pop, q to Quit
1
Enter the element to push: 56
Stack : 34 56
Enter:
1 to Push, 2 to Pop, q to Quit
1
Enter the element to push: 7
Stack : 34 56 7
Enter:
1 to Push, 2 to Pop, q to Quit
2
Stack :
34 56
Enter:
1 to Push, 2 to Pop, q to Quit
1
Enter the element to push: 78
Stack : 34 56 78
Enter:
1 to Push, 2 to Pop, q to Quit
2
Stack :
34 56
Enter:
1 to Push, 2 to Pop, q to Quit
q
```

### Q3) Write a program to reverse a linked list

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

struct Node
{
    int data;
    struct Node *next;
};

void insertLast(struct Node **head,int x)
{
    struct Node* temp=(struct Node*)malloc(sizeof(struct Node));
    temp->data=x;
    temp->next=NULL;
    if(*head==NULL) *head=temp;
    else
    {
        struct Node *p=*head;
        while(p->next)
        {
            p=p->next;
        }
        p->next=temp;
    }
}

struct Node* reverse(struct Node *p)
{
    struct Node *r=NULL,*q=NULL;
    while(p)
    {
        r=q;
        q=p;
        p=p->next;
        q->next=r;
    }
    return q;
}
```



```

void display(struct Node *p)
{
    while(p)
    {
        printf("%d ",p->data);
        p=p->next;
    }
}

int main()
{
    int n=0;
    struct Node *ll1=NULL;
    printf("Enter the No. of elements in the linked list: ");
    scanf("%d",&n);
    printf("Enter the elements seperated by spaces: ");
    while(n>0)
    {
        int num=0;
        scanf("%d",&num);
        insertLast(&ll1,num);
        n--;
    }
    printf("The entered linkedList before reversal: ");
    display(ll1);
    printf("\nThe entered linkedList after reversal: ");
    ll1=reverse(ll1);
    display(ll1);
    return 0;
}

```

## OUTPUT

```

student@HP-Elite600G9-10:~/Desktop/Assignment$ gcc 3_Reverse.c
student@HP-Elite600G9-10:~/Desktop/Assignment$ ./a.out
Enter the No. of elements in the linked list: 8
Enter the elements seperated by spaces: 1 6 8 9 0 4 6 0
The entered linkedList before reversal: 1 6 8 9 0 4 6 0
The entered linkedList after reversal: 0 6 4 0 9 8 6 1 student@HP-Elite600
G9-10:~/Desktop/Assignment$ █

```

Q4 ) Write a function to check if a linked list is palindrome or not?

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

```
struct Node
{
    int data;
    struct Node *next;
};
```

```
void insertLast(struct Node **head,int x)
{
    struct Node* temp=(struct Node*)malloc(sizeof(struct
Node));
    temp->data=x;
    temp->next=NULL;
    if(*head==NULL) *head=temp;
    else
    {
        struct Node *p=*head;
        while(p->next)
        {
            p=p->next;
        }
        p->next=temp;
    }
}
```

```
struct Node* reverse(struct Node *p)
{
    struct Node *r=NULL,*q=NULL;
    while(p)
    {
        r=q;
        q=p;
```

```

        p=p->next;
        q->next=r;
    }
    return q;
}

```

```

struct Node* findMiddle(struct Node*head)
{
    struct Node* fast=head->next;
    struct Node* slow=head;
    while(fast && fast->next)
    {
        slow=slow->next;
        fast=fast->next->next;
    }
    return slow;
}

```

```

void display(struct Node *p)
{
    while(p)
    {
        printf("%d ",p->data);
        p=p->next;
    }
}

```

```

bool isPalindrome(struct Node*head)
{
    struct Node* mid=findMiddle(head);
    struct Node* p=head;
    struct Node* q=mid->next;
    mid->next=NULL;
    q=reverse(q);
    while(p && q)
    {
        if(p->data!=q->data)
        {
            return false;
        }
        p=p->next;

```

```

        q=q->next;
    }
    return true;
}

int main()
{
    int n=0;
    int key=0;
    struct Node *ll1=NULL;
    printf("Enter the No. of elements in the linked list: ");
    scanf("%d",&n);
    printf("Enter the elements seperated by spaces: ");
    while(n>0)
    {
        int num=0;
        scanf("%d",&num);
        insertLast(&ll1,num);
        n--;
    }
    printf("The entered linkedList: ");
    display(ll1);
    if(isPalindrome(ll1)) printf("\nIt is a Palindrome.");
    else printf("\nIt is not a Palindrome.");
    return 0;
}

```

## OUTPUT

```

student@HP-Elite600G9-10:~/Desktop/Assignment$ gcc 4_Palindrome.c
student@HP-Elite600G9-10:~/Desktop/Assignment$ ./a.out
Enter the No. of elements in the linked list: 5
Enter the elements seperated by spaces: 1 2 3 2 1
The entered linkedList: 1 2 3 2 1
It is a Palindrome.student@HP-Elite600G9-10:~/Desktop/Assignment$ 

```

```

student@HP-Elite600G9-10:~/Desktop/Assignment$ gcc 4_Palindrome.c
student@HP-Elite600G9-10:~/Desktop/Assignment$ ./a.out
Enter the No. of elements in the linked list: 6
Enter the elements seperated by spaces: 1 2 3 7 3 2
The entered linkedList: 1 2 3 7 3 2
It is not a Palindrome.student@HP-Elite600G9-10:~/Desktop/Assignment$ 

```