

Coure Name: Data Structures and Algorithms(CBS1003)

ASSESSMENT-1

24BBS0202

ANIRUTH KARTHIKEYAN G

Qn.1)

PSEUDOCODE:

START

Initialize stack[MAX] and top = -1

FUNCTION PUSH():

IF top == MAX - 1:

PRINT "Stack Overflow"

ELSE:

READ element

INCREMENT top by 1

stack[top] = element

FUNCTION POP():

IF top == -1:

PRINT "Stack Underflow"

ELSE:

DECREMENT top by 1

FUNCTION DISPLAY():

IF top == -1:

```
PRINT "Stack is empty"
ELSE:
FOR i = top TO 0:
PRINT stack[i]
MAIN PROGRAM:
WHILE TRUE:
PRINT "1. PUSH 2. POP 3. DISPLAY 4. EXIT"
READ choice
SWITCH(choice):
CASE 1: CALL PUSH()
CASE 2: CALL POP()
CASE 3: CALL DISPLAY()
CASE 4: EXIT
DEFAULT: PRINT "Invalid choice"
END
```

PROGRAM:

```
#include <stdio.h>
#include <stdlib.h>
#define MAX 5
int stack[MAX], top = -1;
void push() {
int element;
if (top == MAX - 1) {
printf("Stack Overflow!\n");
```

```
} else {  
    scanf("%d", &element);  
    stack[++top] = element;  
}  
  
void pop() {  
    if (top == -1) {  
        printf("Stack Underflow!\n");  
    } else {  
        top--;  
    }  
}  
  
void display() {  
    if (top == -1) {  
        printf("Stack is empty.\n");  
    } else {  
        for (int i = top; i >= 0; i--) {  
            printf("%d ", stack[i]);  
        }  
        printf("\n");  
    }  
}  
  
int main() {  
    int choice;  
    while (1) {  
        scanf("%d", &choice);
```

```

switch (choice) {
case 1: push(); break;
case 2: pop(); break;
case 3: display(); break;
case 4: exit(0);
}
}

return 0;
}

```

OUTPUT:

3 TESTCASES:

```

PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-erg5aol5.01p' '--stdout=Microsoft-MIEngine-Out-e1cklany.faj' '--stderr=Microsoft-MIEngine-Error-vn0mxbaz.gf3' '--pid=Microsoft-MIEngine-Pid-mh2ma13f.adj' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter your choice:1
Enter element to be added:3
Enter your choice:1
Enter element to be added:5
Enter your choice:1
Enter element to be added:8
Enter your choice:3
8 5 3
Enter your choice:2
Enter your choice:3
5 3
Enter your choice:4
PS C:\Users\Aniruth Karthikeyan> ^C
PS C:\Users\Aniruth Karthikeyan>
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-pufhvzej.yo0' '--stdout=Microsoft-MIEngine-Out-txfva0jr.fsx' '--stderr=Microsoft-MIEngine-Error-db555iip.nxg' '--pid=Microsoft-MIEngine-Pid-t1dzfbgm.gna' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter your choice:1
Enter element to be added:23
Enter your choice:1
Enter element to be added:12
Enter your choice:3
12 23
Enter your choice:2
Enter your choice:3
23
Enter your choice:4
PS C:\Users\Aniruth Karthikeyan> ^C
PS C:\Users\Aniruth Karthikeyan>
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-r3awglisz.njl' '--stdout=Microsoft-MIEngine-Out-umojuzxx.sgu' '--stderr=Microsoft-MIEngine-Error-hkik30qk.umm' '--pid=Microsoft-MIEngine-Pid-bf104zft.dai' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter your choice:1
Enter element to be added:2
Enter your choice:2
Enter your choice:2
Stack Underflow!

```

Qn.2)

PSEUDOCODE:

START

Initialize queue[MAX], front = -1, rear = -1

FUNCTION ENQUEUE():

IF rear == MAX - 1:

PRINT "Queue Overflow"

ELSE:

READ element

IF front == -1:

front = 0

INCREMENT rear by 1

queue[rear] = element

FUNCTION DEQUEUE():

IF front == -1 OR front > rear:

PRINT "Queue Underflow"

front = -1, rear = -1

ELSE:

INCREMENT front by 1

FUNCTION DISPLAY():

IF front == -1 OR front > rear:

PRINT "Queue is empty"

ELSE:

FOR i = front TO rear:

PRINT queue[i]

MAIN PROGRAM:

```
WHILE TRUE:
PRINT "1. ENQUEUE 2. DEQUEUE 3. DISPLAY 4. EXIT"
READ choice
SWITCH(choice):
CASE 1: CALL ENQUEUE()
CASE 2: CALL DEQUEUE()
CASE 3: CALL DISPLAY()
CASE 4: EXIT
END
```

PROGRAM:

```
#include <stdio.h>

#define n 5

int queue[n];
int front=-1;
int rear=-1;

int IsQueuefull(){
    if (rear==n-1){
        return 1;
    }
    else {
        return 0;
    }
}

void Enqueue(int x){
```

```
if (IsQueuefull()){
    printf("Queue is Full \n");
}
else{
    if (front==-1){
        front+=1;
    }
    rear+=1;
    queue[rear]=x;
}
}

int IsQueueempty(){
    if (front==-1 || front>rear){
        return 1;
    }
    else {
        return 0;
    }
}

void Dequeue(){
    int temp;
    if (IsQueueempty()){
        printf("Queue is empty \n");
    }
    else {
        temp=queue[front];
```

```

        front+=1;
        printf("%d \n",temp);
    }
}

void Display(){
    if (IsQueueempty()){
        printf("Queue is empty \n");
    }
    else {
        for (int i=front ; i<=rear; i++){
            printf("%d \n",queue[i]);
        }
    }
}

int main() {
    int i=1;
    while (i==1)
    {
        int y;
        printf("Enter Choice(1-Push , 2-Pop , 3-Display , 4-stop):" );
        scanf("%d",&y);
        switch (y)
        {
            case 1:
                int a;
                printf("Enter Element:");

```



```
        scanf("%d",&a);
        Enqueue(a);
        break;
case 2:
        Dequeue();
        break;
case 3:
        Display();
        break;
case 4:
        i=0;
        break;
default :
        printf("Enter Valid Choice \n");
        break;
    }
}

return 0;
}
```

OUTPUT:

3 TESTCASES:

```
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-cspinbps.5yr' '--stdout=Microsoft-MIEngine-Out-u1m50w2y.zyv' '--stderr=Microsoft-MIEngine-Error-qc1x4wyu.gk1' '--pid=Microsoft-MIEngine-Pid-whhezaw.o3c' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:23
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:2
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):2
23
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):4
PS C:\Users\Aniruth Karthikeyan> ^C
PS C:\Users\Aniruth Karthikeyan>
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-5zrpmild.sj1' '--stdout=Microsoft-MIEngine-Out-x12ggqw0.ooj' '--stderr=Microsoft-MIEngine-Error-0ywbeuax.vpz' '--pid=Microsoft-MIEngine-Pid-2wdxecih.no4' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:3
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):2
3
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):2
Queue is empty
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):4
PS C:\Users\Aniruth Karthikeyan> ^C
PS C:\Users\Aniruth Karthikeyan>
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-gazjmmli.n2d' '--stdout=Microsoft-MIEngine-Out-4sw3hbai.mij' '--stderr=Microsoft-MIEngine-Error-zafesy3n.iyf' '--pid=Microsoft-MIEngine-Pid-0gbt2yz5.2ur' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:2
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:3
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:4
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:4
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:6
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):1
Enter Element:1
Queue is Full
Enter Choice(1-Enqueue , 2-Dequeue , 3-Display , 4-Exit):3
2
3
4
4
6
```

Ln 51, Col 1 Spaces: 4 UTF-8 CRLF

Qn.3)

PSEUDOCODE:

START

Initialize queue[MAX], front = -1, rear = -1

FUNCTION ENQUEUE():

IF (front == 0 AND rear == MAX - 1) OR (rear + 1) % MAX == front:

PRINT "Queue Overflow"

ELSE:

READ element

IF front == -1:

```
front = 0
rear = (rear + 1) % MAX
queue[rear] = element
FUNCTION DEQUEUE():
    IF front == -1:
        PRINT "Queue Underflow"
    ELSE:
        IF front == rear:
            front = -1, rear = -1
        ELSE:
            front = (front + 1) % MAX
FUNCTION DISPLAY():
    IF front == -1:
        PRINT "Queue is empty"
    ELSE:
        SET i = front
        WHILE i != rear:
            PRINT queue[i]
            i = (i + 1) % MAX
        PRINT queue[rear]
MAIN PROGRAM:
    WHILE TRUE:
        PRINT "1. ENQUEUE 2. DEQUEUE 3. DISPLAY 4. EXIT"
        READ choice
        SWITCH(choice):
            CASE 1: CALL ENQUEUE()
```

CASE 2: CALL DEQUEUE()

CASE 3: CALL DISPLAY()

CASE 4: EXIT

END

PROGRAM:

```
#include <stdio.h>

#include <stdlib.h>

#define N 5

int queue[N], front = -1, rear = -1;

void enqueue(int x) {
    if(front==(rear+1)%N){
        printf("Queue is full");
    }
    else if(front== -1 && rear== -1){
        front=rear=0;
        queue[rear]=x;
    }
    else{
        rear=(rear+1)%N;
        queue[rear]=x;
    }
}

void dequeue(){
    if(front== -1 && rear== -1){
```

```

        printf("Queue is empty\n");
    }
    else if(front==rear){
        front=rear=-1;
    }
    else{
        front=(front+1)%N;
    }
}

void Display(){
    int i=front;
    if (front==-1 && rear==-1){
        printf("Queue is empty\n");
    }
    else {
        while(i!=rear){
            printf("%d ",queue[i]);
            i=(i+1)%N;
        }
        printf("%d \n",queue[i]);
    }
}

int main() {
    int i=1;
    while (i==1)
    {

```

```
int y;
printf("Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):" );
scanf("%d",&y);
switch (y)
{
    case 1:
        int a;
        printf("Enter Element:");
        scanf("%d",&a);
        enqueue(a);
        break;
    case 2:
        dequeue();
        break;
    case 3:
        Display();
        break;
    case 4:
        i=0;
        break;
    default :
        printf("Enter Valid Choice \n");
        break;
}
```

```

    return 0;
}

```

OUTPUT:

3 TESTCASES:

```

PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLaunch
.exe' '--stdin=Microsoft-MIEngine-In-egztxtav.14c' '--stdout=Microsoft-MIEngine-Out-fwoqh3nu.hgv' '--stderr=Microsoft-MIEngine-Error-30wb2tuy.v5e' '--pid=Micro
oft-MIEngine-Pid-ukqv0f2c.xep' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:23
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:32
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):3
23 32
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):4
PS C:\Users\Aniruth Karthikeyan> ^C
PS C:\Users\Aniruth Karthikeyan>
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLaunch
.exe' '--stdin=Microsoft-MIEngine-In-4os05fdf.tuq' '--stdout=Microsoft-MIEngine-Out-zudmjtmx.3um' '--stderr=Microsoft-MIEngine-Error-zbhke0ks.zkq' '--pid=Micro
oft-MIEngine-Pid-3ij0hxl1.1ou' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:2
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:5
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):2
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):2
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):2
Queue is emptyEnter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):4
PS C:\Users\Aniruth Karthikeyan> ^C
PS C:\Users\Aniruth Karthikeyan>
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLaunch
.exe' '--stdin=Microsoft-MIEngine-In-atvj22hy.tps' '--stdout=Microsoft-MIEngine-Out-eyv4325w.31j' '--stderr=Microsoft-MIEngine-Error-3xjslprk.j3r' '--pid=Micro
oft-MIEngine-Pid-wbbomwtm.c0l' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:1
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:2
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:3
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:4
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:5
Enter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):1
Enter Element:5
Queue is fullEnter Choice(1-Enqueue ,2-Dequeue ,3-Display ,4-Exit):
PS C:\Users\Aniruth Karthikeyan> █

```

Qn.4)

PSEUDOCODE:

Function insert_b():

Input num

Create a new node t

t.data = num

t.next = NULL

If start is NULL:

 // If the list is empty

 start = t

Else:

 q = start

 While q.next is not NULL:

 q = q.next

 End While

 q.next = t

End If

End Function

Function insert_e():

 Input num

 Create a new node t

 t.data = num

 t.next = NULL

If start is NULL:

 // If the list is empty

 start = t

Else:

 q = start

 While q.next is not NULL:

 q = q.next

 End While

q.next = t

End If

End Function

Function InsertAtEnd(data)

Create newNode

Set newNode.data to data

Set newNode.next to NULL

Set newNode.prev to NULL

If head is NULL

Set head to newNode

Return

Set temp to head

While temp.next is not NULL

Set temp to temp.next

Set temp.next to newNode

Set newNode.prev to temp

End Function

Function insert_p(p, n):

Input: position p, data n

If start is NULL:

Print "List is empty"

Return 0

Create a new node t

t.data = n

q = start

For i = 0 to p - 1:

 If q.next is NULL:

 Print "There are fewer elements"

 Return 0

 End If

 q = q.next

End For

t.next = q.next

q.next = t

Return 0

End Function

Function delete_b():

 If start is NULL:

 Print "The list is empty"

 Else:

 q = start

 start = start.next

 Print "Deleted element is", q.data

 Free q

 End If

End Function

Function delete_e():

 If start is NULL:

 Print "The list is empty"

 Return

 q = start

 While q.next.next is not NULL:

 q = q.next

 End While

 t = q.next

 q.next = NULL

 Print "Deleted element is", t.data

 Free t

End Function

Function delete_p(pos):

 Input: Position pos

 If start is NULL:

 Print "List is empty"

 Return 0

 q = start

 For i = 1 to pos - 1:

 If q.next is NULL:

 Print "There are fewer elements"

Return 0

End If

q = q.next

End For

t = q.next

q.next = t.next

Print "Deleted element is", t.data

Free t

Return 0

End Function

Function search(k):

Input: Key k

flag = 0

temp = start

While temp is not NULL:

If k == temp.data:

flag = 1

Break

End If

temp = temp.next

End While

If flag == 1:

 Print "Key Found"

Else:

 Print "Key Not Found"

End If

End Function

Main Function:

 Loop forever:

 Print "Enter Choice"

 Read choice

 Switch choice:

 Case 1:

 Print "Enter data to insert at the beginning"

 Read data

 Call insertBeginning(data)

 Case 2:

 Print "Enter data to insert at the end"

 Read data

 Call insertEnd(data)

 Case 3:

 Print "Enter data to insert"

 Read data

 Print "Enter position to insert at"

 Read position

 Call insertAtPosition(data, position)

 Case 4:

Call deleteBeginning()

Case 5:

Call deleteEnd()

Case 6:

Print "Enter position to delete from"

Read position

Call deleteAtPosition(position)

Case 7:

Print "Enter element to search for"

Read data

Call search(data)

Case 9:

Print "Exiting program"

Exit

Default:

Print "Invalid choice. Please try again"

PROGRAM:

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

```
#include<conio.h>
```

```
#include<process.h>
```

```
struct node
```

```
{
```

```

    int data;

    struct node *next;
}*start=NULL,*q,*t;

int main()
{
    int ch;

    void insert_beg();
    void insert_end();
    int insert_pos();
    void display();
    void delete_beg();
    void delete_end();
    int delete_pos();

    while(1)
    {
        printf("\n\n---- Singly Linked List(SLL) Menu ----");
        printf("\n1.Insert\n2.Display\n3.Delete\n4.Exit\n\n");
        printf("Enter your choice(1-4):");
        scanf("%d",&ch);

        switch(ch)
        {
            case 1:
                printf("\n---- Insert Menu ----");

```

```
printf("\n1.Insert at beginning\n2.Insert at end\n3.Insert at  
specified position\n4.Exit");
```

```
printf("\n\nEnter your choice(1-4):");
```

```
scanf("%d",&ch);
```

```
switch(ch)
```

```
{
```

```
case 1: insert_beg();
```

```
break;
```

```
case 2: insert_end();
```

```
break;
```

```
case 3: insert_pos();
```

```
break;
```

```
case 4: exit(0);
```

```
default: printf("Wrong Choice!!");
```

```
}
```

```
break;
```

```
case 2: display();
```

```
break;
```

```
case 3: printf("\n---- Delete Menu ----");
```

```
printf("\n1.Delete from beginning\n2.Delete from end\n3.Delete  
from specified position\n4.Exit");
```

```
printf("\n\nEnter your choice(1-4):");
```

```
scanf("%d",&ch);
```



```

switch(ch)
{
    case 1: delete_beg();
            break;
    case 2: delete_end();
            break;
    case 3: delete_pos();
            break;
    case 4: exit(0);
    default: printf("Wrong Choice!!");
}
break;
case 4: exit(0);
default: printf("Wrong Choice!!");
}
}
return 0;
}

```

```

void insert_beg()
{
    int num;
    t=(struct node*)malloc(sizeof(struct node));
    printf("Enter data:");
    scanf("%d",&num);
    t->data=num;
}

```

```
if(start==NULL)    //If list is empty
{
    t->next=NULL;
    start=t;
}
else
{
    t->next=start;
    start=t;
}
}
```

```
void insert_end()
{
    int num;
    t=(struct node*)malloc(sizeof(struct node));
    printf("Enter data:");
    scanf("%d",&num);
    t->data=num;
    t->next=NULL;

    if(start==NULL)    //If list is empty
    {
        start=t;
    }
}
```

```
else
{
    q=start;
    while(q->next!=NULL){
        q=q->next;
        q->next=t;
    }
}
```

```
int insert_pos()
{
    int pos,i,num;
    if(start==NULL)
    {
        printf("List is empty!!");
        return 0;
    }
```

```
t=(struct node*)malloc(sizeof(struct node));
printf("Enter data:");
scanf("%d",&num);
printf("Enter position to insert:");
scanf("%d",&pos);
t->data=num;
```

```
q=start;
for(i=0;i<pos;i++)
{
    if(q->next==NULL)
    {
        printf("There are less elements!!");
        return 0;
    }

    q=q->next;
}

t->next=q->next;
q->next=t;
return 0;
}

void display()
{
    if(start==NULL)
    {
        printf("List is empty!!");
    }
    else
    {
        q=start;
```

```
    printf("The linked list is:\n");
    while(q!=NULL)
    {
        printf("%d->",q->data);
        q=q->next;
    }
}
```

```
void delete_beg()
{
    if(start==NULL)
    {
        printf("The list is empty!!");
    }
    else
    {
        q=start;
        start=start->next;
        printf("Deleted element is %d",q->data);
        free(q);
    }
}
```

```
void delete_end()
{
```

```
if(start==NULL)
{
    printf("The list is empty!!");
}
else
{
    q=start;
    while(q->next->next!=NULL)
    q=q->next;

    t=q->next;
    q->next=NULL;
    printf("Deleted element is %d",t->data);
    free(t);
}
}
```

```
int delete_pos()
{
    int pos,i;

    if(start==NULL)
    {
        printf("List is empty!!");
        return 0;
    }
```

```
printf("Enter position to delete:");
```

```
scanf("%d",&pos);
```

```
q=start;
```

```
for(i=0;i<pos;i++)
```

```
{
```

```
    if(q->next==NULL)
```

```
    {
```

```
        printf("There are less elements!!");
```

```
        return 0;
```

```
    }
```

```
    q=q->next;
```

```
}
```

```
t=q->next;
```

```
q->next=t->next;
```

```
printf("Deleted element is %d",t->data);
```

```
free(t);
```

```
return 0;
```

```
}
```

OUTPUT:

```
PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-t0erm3z.u11' '--stdout=Microsoft-MIEngine-Out-v35k2qvn.o0s' '--stderr=Microsoft-MIEngine-Error-qmojaww5.uh0' '--pid=Microsoft-MIEngine-Pid-n015ziug.tgu' '--dbgExe=c:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
```

```
---- Singly Linked List(SLL) Menu ----
1.Insert
2.Display
3.Delete
4.Exit
```

Enter your choice(1-4):1

```
---- Insert Menu ----
1.Insert at beginning
2.Insert at end
3.Insert at specified position
4.Exit
```

Enter your choice(1-4):1
Enter data:34

```
---- Singly Linked List(SLL) Menu ----
1.Insert
2.Display
3.Delete
4.Exit
```

Enter your choice(1-4):1

```
---- Insert Menu ----
1.Insert at beginning
2.Insert at end
3.Insert at specified position
4.Exit
```

Enter your choice(1-4):2
Enter data:34

```
---- Singly Linked List(SLL) Menu ----
1.Insert
2.Display
3.Delete
4.Exit
```

Enter your choice(1-4):1

```
---- Insert Menu ----
1.Insert at beginning
2.Insert at end
3.Insert at specified position
4.Exit
```

Enter your choice(1-4):3
Enter data:11
Enter position to insert:2
There are less elements!!

```
---- Singly Linked List(SLL) Menu ----
1.Insert
2.Display
3.Delete
4.Exit
```

Enter your choice(1-4):2
The linked list is:
34->34->

```
---- Singly Linked List(SLL) Menu ----
1.Insert
2.Display
3.Delete
4.Exit
```

Enter your choice(1-4):4
PS C:\Users\Aniruth Karthikeyan> █

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):2

The linked list is:

23->54->32->34->

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):3

---- Delete Menu ----

- 1.Delete from beginning
- 2.Delete from end
- 3.Delete from specified position
- 4.Exit

Enter your choice(1-4):1

Deleted element is 23

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):2

The linked list is:

2->5->3->

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):3

---- Delete Menu ----

- 1.Delete from beginning
- 2.Delete from end
- 3.Delete from specified position
- 4.Exit

Enter your choice(1-4):2

Deleted element is 3

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):1

---- Insert Menu ----

- 1.Insert at beginning
- 2.Insert at end
- 3.Insert at specified position
- 4.Exit

Enter your choice(1-4):1

Enter data:4

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):3

---- Delete Menu ----

- 1.Delete from beginning
- 2.Delete from end
- 3.Delete from specified position
- 4.Exit

Enter your choice(1-4):3

Enter position to delete:1

Deleted element is 1

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):2

The linked list is:

12->32->23->2->5->

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):3

---- Delete Menu ----

- 1.Delete from beginning
- 2.Delete from end
- 3.Delete from specified position
- 4.Exit

Enter your choice(1-4):2

Deleted element is 5

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):3

---- Delete Menu ----

- 1.Delete from beginning
- 2.Delete from end
- 3.Delete from specified position
- 4.Exit

Enter your choice(1-4):1

Deleted element is 12

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Enter your choice(1-4):2

The linked list is:

32->23->2->

---- Singly Linked List(SLL) Menu ----

- 1.Insert
- 2.Display
- 3.Delete
- 4.Exit

Qn.5)

PSEUDOCODE:

Define a structure Node:

Integer data

Pointer to Node prev

Pointer to Node next

Initialize head as NULL

Function insertBeginning(data):

Create a new node

Set newNode->data = data

Set newNode->prev = NULL

Set newNode->next = head

If head is not NULL:

Set head->prev = newNode

Set head = newNode

Function insertEnd(data):

Create a new node

Set newNode->data = data

Set newNode->next = NULL

If head is NULL:

Set newNode->prev = NULL

Set head = newNode

Return

Set temp = head

While temp->next is not NULL:

Set temp = temp->next

Set temp->next = newNode

Set newNode->prev = temp

Function insertAtPosition(data, position):

If position <= 0:

Print "Invalid position"

Return

Create a new node

Set newNode->data = data

If position is 1:

Set newNode->prev = NULL

Set newNode->next = head

If head is not NULL:

Set head->prev = newNode

Set head = newNode

Return

Set temp = head

For i = 1 to position - 1:

If temp is NULL:

Print "Position exceeds the size of the list"

Free newNode

Return

Set temp = temp->next

If temp is NULL:

Print "Position exceeds the size of the list"

Free newNode

Return

Set newNode->next = temp->next

Set newNode->prev = temp

If temp->next is not NULL:

Set temp->next->prev = newNode

Set temp->next = newNode

Function deleteBeginning():

If head is NULL:

Print "The list is empty"

Return

Set temp = head

Set head = head->next

If head is not NULL:

Set head->prev = NULL

Print "Deleted element is temp->data"

Free temp

Function deleteEnd():

If head is NULL:

Print "The list is empty"

Return

Set temp = head

If head->next is NULL:

Set head = NULL

Print "Deleted element is temp->data"

Free temp

Return

While temp->next is not NULL:

Set temp = temp->next

Set temp->prev->next = NULL

Print "Deleted element is temp->data"

Free temp

Function deleteAtPosition(position):

If head is NULL:

Print "The list is empty"

Return

If position <= 0:

Print "Invalid position"

Return

Set temp = head

If position is 1:

Set head = head->next

If head is not NULL:

Set head->prev = NULL

Print "Deleted element is temp->data"

Free temp

Return

For i = 1 to position:

If temp is NULL:

Print "Position exceeds the size of the list"

Return

Set temp = temp->next

If temp is NULL:

Print "Position exceeds the size of the list"

Return

If temp->next is not NULL:

Set temp->next->prev = temp->prev

If temp->prev is not NULL:

Set temp->prev->next = temp->next

Print "Deleted element is temp->data"

Free temp

Function search(key):

Set temp = head

Set flag = 0

While temp is not NULL:

If temp->data is equal to key:

Set flag = 1

Set temp = temp->next

If flag is 1:

Print "Element found"

Else:

Print "Element not found"

Main Function:

Loop forever:

Print "Enter Choice"

Read choice

Switch choice:

Case 1:

Print "Enter data to insert at the beginning"

Read data

Call insertBeginning(data)

Case 2:

Print "Enter data to insert at the end"

Read data

Call insertEnd(data)

Case 3:

Print "Enter data to insert"

Read data

Print "Enter position to insert at"

Read position

Call insertAtPosition(data, position)

Case 4:

Call deleteBeginning()

Case 5:

Call deleteEnd()

Case 6:

Print "Enter position to delete from"

Read position

Call deleteAtPosition(position)

Case 7:

Print "Enter element to search for"

Read data

Call search(data)

Case 9:

Print "Exiting program"

Exit

Default:

Print "Invalid choice. Please try again"

PROGRAM:

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

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

```
struct Node {
```

```
    int data;
```

```
    struct Node *prev;
```

```
    struct Node *next;
```

```
};
```

```
struct Node *head = NULL;
```

```
void insertBeginning(int data) {
```

```
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
```

```
    newNode->data = data;
```

```
    newNode->prev = NULL;
```

```
    newNode->next = head;
```

```
    if (head != NULL) {
```

```
        head->prev = newNode;
```

```
    }
```

```
    head = newNode;
```

```
}
```

```
void insertEnd(int data) {
```

```
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
```

```
    struct Node *temp = head;
```

```

newNode->data = data;
newNode->next = NULL;
if (head == NULL) {
    newNode->prev = NULL;
    head = newNode;
    printf("Node inserted at the end.\n");
    return;
}
while (temp->next != NULL) {
    temp = temp->next;
}
temp->next = newNode;
newNode->prev = temp;
}

```

```

void insertAtPosition(int data, int position) {
    if (position <= 0) {
        printf("Invalid position.\n");
        return;
    }
    struct Node *newNode = (struct Node *)malloc(sizeof(struct Node));
    struct Node *temp = head;
    int i;
    newNode->data = data;
    if (position == 1) {
        newNode->prev = NULL;

```

```

newNode->next = head;
if (head != NULL) {
    head->prev = newNode;
}
head = newNode;
return;
}
for (i = 1; i < position - 1; i++) {
    if (temp == NULL) {
        printf("Position exceeds the size of the list.\n");
        free(newNode);
        return;
    }
    temp = temp->next;
}
if (temp == NULL) {
    printf("Position exceeds the size of the list.\n");
    free(newNode);
    return;
}
newNode->next = temp->next;
newNode->prev = temp;
if (temp->next != NULL) {
    temp->next->prev = newNode;
}
temp->next = newNode;

```

```
    printf("Node inserted at position %d.\n", position);  
}
```

```
void deleteBeginning() {  
    if (head == NULL) {  
        printf("The list is empty.\n");  
        return;  
    }  
    struct Node *temp = head;  
    head = head->next;  
    if (head != NULL) {  
        head->prev = NULL;  
    }  
    printf("Deleted element is %d.\n", temp->data);  
    free(temp);  
}
```

```
void deleteEnd() {  
    if (head == NULL) {  
        printf("The list is empty.\n");  
        return;  
    }  
    struct Node *temp = head;  
    if (head->next == NULL) {  
        head = NULL;  
        printf("Deleted element is %d.\n", temp->data);  
    }  
}
```

```
    free(temp);
    return;
}
while (temp->next != NULL) {
    temp = temp->next;
}
temp->prev->next = NULL;
printf("Deleted element is %d.\n", temp->data);
free(temp);
}
```

```
void deleteAtPosition(int position) {
    if (head == NULL) {
        printf("The list is empty.\n");
        return;
    }
    if (position <= 0) {
        printf("Invalid position.\n");
        return;
    }
    struct Node *temp = head;
    int i;
    if (position == 1) {
        head = head->next;
        if (head != NULL) {
            head->prev = NULL;
        }
    }
}
```

```

    }
    printf("Deleted element is %d.\n", temp->data);
    free(temp);
    return;
}
for (i = 1; i < position; i++) {
    if (temp == NULL) {
        printf("Position exceeds the size of the list.\n");
        return;
    }
    temp = temp->next;
}
if (temp == NULL) {
    printf("Position exceeds the size of the list.\n");
    return;
}
if (temp->next != NULL) {
    temp->next->prev = temp->prev;
}
if (temp->prev != NULL) {
    temp->prev->next = temp->next;
}
printf("Deleted element is %d.\n", temp->data);
free(temp);
}

```



```

void search(int key) {
    struct Node *temp = head;
    int flag=0;
    while (temp != NULL) {
        if (temp->data == key) {
            flag=1;
        }
        temp = temp->next;
    }
    if (flag==1){
        printf("Element found .\n");
    }
    else{
        printf("Element not found .");
    }
}

```

```

int main() {
    int choice, data, position;
    while (1) {
        printf("Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos  
7-Search , 8-END):");
        scanf("%d", &choice);
        switch (choice) {
            case 1:
                printf("Enter data to insert at the beginning: ");
                scanf("%d", &data);

```

```
    insertBeginning(data);  
    break;  
case 2:  
    printf("Enter data to insert at the end: ");  
    scanf("%d", &data);  
    insertEnd(data);  
    break;  
case 3:  
    printf("Enter data to insert: ");  
    scanf("%d", &data);  
    printf("Enter position to insert at: ");  
    scanf("%d", &position);  
    insertAtPosition(data, position);  
    break;  
case 4:  
    deleteBeginning();  
    break;  
case 5:  
    deleteEnd();  
    break;  
case 6:  
    printf("Enter position to delete from: ");  
    scanf("%d", &position);  
    deleteAtPosition(position);  
    break;  
case 7:
```

```

        printf("Enter element to search for: ");

        scanf("%d", &data);

        search(data);

        break;

case 9:

    printf("Exiting program.\n");

    exit(0);

default:

    printf("Invalid choice. Please try again.\n");

    }

}

return 0;

}

```

OUTPUT:

3 TESTCASES:

```

PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64
.exe' '--stdin=Microsoft-MIEngine-In-zmd31f0k.yow' '--stdout=Microsoft-MIEngine-Out-q32u4k0c.wh2' '--stderr=Microsoft-MIE
oft-MIEngine-Pid-1idiltpe.hxd' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 2
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):2
Enter data to insert at the end: 4
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):3
Enter data to insert: 3
Enter position to insert at: 1
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):4
Deleted element is 3.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):5
Deleted element is 4.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):6
Enter position to delete from: 1
Deleted element is 2.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):8
Invalid choice. Please try again.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):

```

```

Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 2
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 4
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 5
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):3
Enter data to insert: 7
Enter position to insert at: 2
Node inserted at position 2.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):2
Enter data to insert at the end: 8
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):2
Enter data to insert at the end: 4
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):7
Enter element to search for: 7
Element found .
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):7
Enter element to search for: 2
Element found .
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):7
Enter element to search for: 9
Element not found .Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):4
Deleted element is 5.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):5
Deleted element is 4.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):6
Enter position to delete from: 2
Deleted element is 4.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):

```

```

PS C:\Users\Aniruth Karthikeyan> & 'c:\Users\Aniruth Karthikeyan\.vscode\extensions\ms-vscode.cpptools-1.22.11-win32-x64\debugAdapters\bin\WindowsDebugLauncher.exe' '--stdin=Microsoft-MIEngine-In-keggj0yf.fkl' '--stdout=Microsoft-MIEngine-Out-yqkatpyt.c2a' '--stderr=Microsoft-MIEngine-Error-1kkvt4yy.t41' '--pid=Microsoft-MIEngine-Pid-i5n5wxs1.23m' '--dbgExe=C:\msys64\ucrt64\bin\gdb.exe' '--interpreter=mi'
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 3
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 65
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):1
Enter data to insert at the beginning: 2
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):2
Enter data to insert at the end: 63
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):3
Enter data to insert: 34
Enter position to insert at: 3
Node inserted at position 3.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):7
Enter element to search for: 45
Element not found .Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):7
Enter element to search for: 63
Element found .
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):4
Deleted element is 2.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):5
Deleted element is 63.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):6
Enter position to delete from: 3
Deleted element is 3.
Enter Choice (Insert: 1-Beg 2-End 3-Pos Deletion: 4-Beg 5-End 6-Pos 7-Search , 8-END):8
Exiting program.
PS C:\Users\Aniruth Karthikeyan>

```