

ASSIGNMENT - 4

C program to realize singly linked list:

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int val;
    struct node *next;
};
```

```
void printlist(struct node *head)
```

```
{ printf("%d->", head->val);
```

```
while(head)
```

```
{
```

```
    printf("%d->", head->val);
```

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

```
}
```

```
}
```

```
void insertfront(struct node **head, int value)
```

```
{
```

```
    struct node *newnode = new;
```

```
    newnode = (struct node *) malloc(sizeof  
        (struct node));
```

```
    if(newnode == NULL)
```

```
{
```

```
        printf("Failed to insert");
```

```
}
```

```
    newnode->val = value;
```

```
    newnode->next = *head;
```

```
*head = newnode;
```

```
3
```

Teacher's Signature

Scanned by TapScanner

```
void main()
```

{

```
int count = 0, i, val;
```

```
struct node *head = null;
```

```
printf("Enter no of elements");
```

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

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

{

```
printf("Enter the %dth element", i);
```

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

```
insertfront(&head, val);
```

3

```
printf("linked list");
```

```
printList(head);
```

3

C prog to implement Stack using linked list

#include < stdio.h >

#include < stdlib.h >

struct node

{

int data;

struct node *next;

};

void initialize()

{

top = NULL;

}

int isEmpty()

{ if (top == NULL)

return 1;

else

return 0;

}

int peek()

{

return top->data;

}

Teacher's Signature _____

int getSize (struct node *head)

{

} if (head == NULL)

{

printf ("Invalid stack pointer")
return;

3

int length = 0;

while (head != NULL)

{

head = head->next;
length++;

3

return length;

3

void Push (int num)

{

struct node *temp;

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

temp->data = num;

} if (top == NULL)

{

top = temp;

top->next = NULL;

3

else

{

temp->next = top;

top = temp;

3

3

Teacher's Signature _____

void Pop()

{

Struct node *temp;

if (isEmpty (top))

{ printf ("Empty"); }

return;

}

else

{

temp = top;

top = top->next;

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

free (temp);

}

3

void printStack (Struct node *nodePtr)

{

while (nodePtr != NULL)

{

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

nodePtr = nodePtr->next;

if (nodePtr != NULL)

printf ("-->");

3

printf ("\n");

3

```
void main()
{
    initialize();
    push(1);
    push(2);
    push(3);
    push(4);
    cout << "Stack size " << d[4] << ", getStackSize(" << top << ")";
    cout << "Top Element " << peek() << ",";
    cout << "Stack as linkedlist ";
    printStack(top);
    pop();
    pop();
    pop();
    cout << endl;
    printStack(top);
    return;
}
```

3

Output :

Stack size : 4

Top Element : 4

Stack as linked list

4 → 3 → 2 → 1

Removed element : 4

Removed element : 3

Removed element : 2

Removed element : 1

Stack is Empty.

e) Program to implement Queue Using Singly linked list:

#include < stdio.h >

#include < stdlib.h >

struct node

{

int data;

struct node *next;

};

struct node *front;

struct node *rear;

void ins();

void del();

void display();

void main()

}

int choice;

while (choice != 4)

{

printf ("1. Insert 2. Delete 3. Display 4. Exit 1");

printf ("Enter choice");

scanf ("%d", &choice);

switch (choice)

{

case 1: ins();

break;

case 2: del();

break;

Teacher's Signature _____

case 3 : display();
break;

case 4 : Exit(0);

default : printf ("Invalid choice");

3

3

void ins()

{

Struct node *ptr;

int item;

ptr = (struct node *) malloc (sizeof (struct node));

? (ptr == null)

{

printf ("In overflow");

return;

3

else

{

printf ("Enter value");

scanf ("%d", &item);

ptr->data = item;

? (front == null)

{

front = ptr;

rear = ptr;

front->next = null;

for rear->next = null;

3

Teacher's Signature _____

else

{

rear → next = ptr;

rear = ptr;

rear → next = NULL;

3

3

3

void del()

{

struct node *ptr;

if (front == NULL)

{

printf ("underflow");

return;

3

else

{

ptr = front;

front = front → next;

free(ptr);

3

3

void display()

{

struct node *ptr;

ptr = front;

if (front == NULL)

```
2
printf ("Empty queue");
3
else
{
    printf ("Elements of Queue");
    while (ptr != NULL)
    {
        printf ("%d", ptr->data);
        ptr=ptr->next;
    }
    printf ("\n");
}
3
```

Output:

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

Enter choice : 1

Enter value : 10

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

Enter choice : 1

Enter value : 20

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

Enter choice : 2

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

Enter choice : 3

Element of Queue : 20

C program to implement Queue Using Circular
Linked List:

```
#include <stdio.h>
#include <stdlib.h>
struct node
{
    int info;
    struct node *link;
} *rear = NULL;
```

```
void insert(int item);
int del();
void display();
int isEmpty();
int peek();
int main()
```

```
{
```

```
int choice, item;
```

```
while(1)
```

```
{
```

```
printf("1.Insert 2.Delete 3.Display 4.Exit");
printf("Enter choice");
scanf("%d", &choice);
switch(choice)
```

```
{
```

```
case 1: printf("Enter element to insert");
scanf("%d", &item);
insert(item);
break;
```

Teacher's Signature _____

case 2 : printf ("Deleted element is %d, del();
break;

Case 3 : printf ("Item at front of queue %d", peek());
break;

Case 4 : display();
break;

Case 5 : exit(1);

default : printf ("Wrong choice");

3

3

3

void insert (int item)

{

struct node *tmp;

tmp = (struct node *) malloc (sizeof (struct node));

tmp->info = item;

if (tmp == NULL)

{

printf ("Memory not available");

return;

3

if (is Empty())

{

rear = tmp;

tmp->links = rear;

3

Teacher's Signature _____

else

{

tmp → link = rear → link;

rear → link = tmp;

rear = tmp;

3

del()

{

int item;

struct node *tmp;

} if (IsEmpty())

{

printf (" Queue underflow");

exit();

3

if (rear → link == rear) .

{

temp = rear;

rear = NULL;

3

else

{

tmp = rear → link;

rear → link = rear → link → link;

3

item = temp → info;

free (temp);

return item;

8

Teacher's Signature _____

```
int peek()
{
    if (isEmpty())
        printf("Queue underflow")
        exit(1);
}
```

return rear->link->info;

3 ~~End~~.

```
int isEmpty()
{
    if (rear == NULL)
        return 1;
    else
        return 0;
}
```

```
void display()
{
    struct node *p;
    if (isEmpty())
        printf("Queue is Empty");
    return;
}
```

3
printf(" Queue is")
 $p = rear \rightarrow link$
else

2

```
printf("odd", p->info);
```

```
p=p->link;
```

```
} while(p!=rear->link);
```

3

Output:

1. Insert 2. Display 3. Delete 4. Peek 5. Quit

Enter choice: 1

Enter element to insert: 1

1. Insert 2. Delete 3. Peek 4. Display 5. Quit

Enter choice: 2

Enter element to insert: 2

1. Insert 2. Delete 3. Peek 4. Display 5. Quit

Enter choice: 2

Deleted element is: 1

1. Insert 2. Delete 3. Peek 4. Display 5. Quit

Enter choice: 4

Queue is: 2