

1)Stack Using Linked List

Code:

```
#include<stdio.h>

#include<stdlib.h>

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

int isempty(struct node *top)
{
    if(top==NULL)
        return 1;
    else
        return 0;
}

int isfull()
{
    struct node *n=(struct node *)malloc(sizeof(struct
node));
```

```
if(n==NULL)
return 1;
else
return 0;
}
struct node *push(struct node *top,int val)
{
    struct node *nn=(struct node *)malloc(sizeof(struct
node));
    if(isfull())
    {
        printf("Stack Is Full\n");
    }
    else
    {
        nn->data=val;
        nn->next=top;
        top=nn;
        return top;
    }
}
```

```
};  
  
struct node *pop(struct node *top)  
{  
    if(isempty(top))  
    {  
        printf("Cant Pop Stack Empty\n");  
    }  
    else  
    {  
        struct node *temp=top;  
        top=top->next;  
        printf("Node with data %d",temp->data);  
        printf(" Removed\n");  
        free(temp);  
        return top;  
    }  
};  
  
void display(struct node *top)  
{  
    while(top!=NULL)
```

```
{
    printf("%d\n",top->data);
    top=top->next;
}
}
int main()
{
    struct node *top=NULL;
    int ch;
    int data1;
    do
    {
        printf("Option 1 For Pop\n");
        printf("Option 2 For Push\n");
        printf("Option 3 For Display\n");
        printf("Option 4 Exit\n");
        printf("Enter Choice\n");
        scanf("%d",&ch);
        switch(ch)
        {
```

```
    case 1:
        top=pop(top);
        break;
    case 2:
        printf("Enter Node Data\n");
        scanf("%d",&data1);
        top=push(top,data1);
        break;
    case 3:
        display(top);
        break;
    }
}while(ch!=4);
}
```

Output:

```
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
2
Enter Node Data
10
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
2
Enter Node Data
20
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
2
Enter Node Data
30
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
2
Enter Node Data
40
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
3
40
30
20
10
Option 1 For Pop
```

```
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
1
Node with data 40 Removed
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
3
30
20
10
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
1
Node with data 30 Removed
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
3
20
10
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
2
Enter Node Data
50
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
```

```
3
50
20
10
Option 1 For Pop
Option 2 For Push
Option 3 For Display
Option 4 Exit
Enter Choice
4
```

