

```

#include<stdio.h>
#include<stdlib.h>
typedef struct stack
{
    int data;
    struct stack *next;
}stack;
stack *top=NULL;
void create();
void display();
void push(int x);
int pop();
int peek(int pos);
int stackTop();
int isEmpty();
int isFull();

int main()
{
    int choice,x,i,pos;
    do
    {
        printf("\nEnter choice:\n1.Push\n2.Pop\n3.Display\n4.Peek at any position\n5.Look
at Top position\n6.Check if Empty\n7.Check if Full\n8.Exit\n");
        scanf("%d",&choice);
        switch(choice)
        {
            case 1: printf("\nEnter element you want to push: ");
                    scanf("%d",&x);
                    push(x);
                    break;
            case 2: i=pop();
                    if(i!=-1)
                        printf("\nPopped element is: %d\n",i);
                    else
                        printf("\nStack is Empty\n");
                    break;
            case 3: display();
                    break;
            case 4: printf("Enter position to peek: ");
                    scanf("%d",&pos);
                    if(i=peek(pos))
                        printf("\nElement at position %d in stack is: %d\n",pos,i);
                    break;
            case 5: if(i=stackTop())
                    printf("\nFirst element in stack is: %d\n",i);
                    break;
            case 6: if(isEmpty())
                    printf("\nStack is Empty\n");
                    else
                        printf("\nStack is not Empty\n");
                    break;
            case 7: if(isFull())
                    printf("\nStack is Full\n");
                    else
                        printf("\nStack is not Full\n");
                    break;
            case 8: return 0;
        }
    }while(1);
}

void push(int x)
{
    stack *new_node;
    new_node=(stack *)malloc(sizeof(stack));

```

```

    if(new_node==NULL)
    {
        printf("Stack Overflow condition");
        return;
    }
    new_node->data=x;
    new_node->next=NULL;
    if(top==NULL)
        top=new_node;
    else
    {
        new_node->next=top;
        top=new_node;
    }
}

int pop()
{
    int x=-1;
    stack *ptr=top;
    if(top==NULL)
        return x;
    else
    {
        top=top->next;
        x=ptr->data;
        free(ptr);
    }
    return x;
}

void display()
{
    if(top==NULL)
    {
        printf("\nStack is Empty\n");
        return 0;
    }
    else
    {
        stack *ptr=top;
        printf("\nStack element are: ");
        while(ptr)
        {
            printf("%d ",ptr->data);
            ptr=ptr->next;
        }
    }
}

int peek(int pos)
{
    if(top==NULL)
    {
        printf("\nStack is Empty\n");
        return 0;
    }
    int x=-1,i;
    stack *ptr=top;
    for(i=0;i<pos-1 && ptr!=NULL;i++)
        ptr=ptr->next;
    if(ptr)
        return ptr->data;
    else
    {
        printf("\nPosition is INVALID\n");
    }
}

```

```

        return 0;
    }
}

int stackTop()
{
    if(top==NULL)
    {
        printf("\nStack is Empty\n");
        return 0;
    }
    else
        return top->data;
}

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

int isFull()
{
    stack *new_node;
    new_node=(stack *)malloc(sizeof(stack));
    if(new_node==NULL)
        return 1;
    else
        return 0;
}

```