

Program – Stack ADT

```
#include<iostream>
using namespace std;
class stack
{
    private:
        int a[10];
        int maxcapacity;
        int top;
    public:
        stack()
        {
            maxcapacity=10;
            top=-1;
        }
        void push(int element);
        int pop();
        void display();
};

void stack::push(int element)
{
    if(top==maxcapacity-1)
        cout<<"stack overflow";
    else
        a[++top]=element;
}

int stack::pop()
{
    if(top==-1)
        cout<<"stack underflow";
    else
    {
        cout<<a[top]<<" is popped out";
        return(a[top--]);
    }
}

void stack::display()
{
    cout<<"*****"<<endl;
    if(top==-1)
        cout<<"stack underflow";
    else
```

```

        for(int i=top;i>=0;i--)
            cout<<a[i]<<endl;
        cout<<"*****"<<endl;
    }
    int main()
    {
        int ch,val;
        stack s;
        while(1)
        {
            cout<<"\n1.push\n2.pop\n3.display\n4.exit\n";
            cin>>ch;
            switch(ch)
            {
                case 1: cout<<"enter the element to push:";
                        cin>>val;
                        s.push(val);
                        cout<<val<<"is added to stack"<<endl;
                        break;
                case 2:s.pop();
                        break;
                case 3:s.display();
                        break;
                case 4:exit(0);
            }
        }
    }

```

Program – Queue ADT

```
#include<iostream>
using namespace std;
class queue
{
    private:
        int a[10];
        int maxcapacity;
        int rear;
        int front;
    public:
        queue()
        {
            maxcapacity=10;
            rear=-1;
            front=-1;
        }
        void insert(int element);
        int delet();
        void display();
};
void queue::insert(int element)
{
    if(rear==maxcapacity-1)
        cout<<"queue is full";
    else
    {
        a[++rear]=element;
        front=0;
        cout<<element<<" is added to queue"<<endl;
    }
}
int queue::delet()
{
    if(front==-1 && rear==-1)
        cout<<"Queue is empty";
    else if(front==rear)
    {
        cout<<a[front]<<" is deleted";
        front=-1;
        rear=-1;
    }
}
```

```

    }
    else
    {
        cout<<a[front]<<" is deleted";
        a[front++];
    }
}
void queue::display()
{
    cout<<"*****"<<endl;
    if(front==-1 && rear==-1)
        cout<<"queue is empty";
    else
        for(int i=front;i<=rear;i++)
            cout<<a[i]<<endl;
    cout<<"*****"<<endl;
}
int main()
{
    int ch,val;
    queue q;
    while(1)
    {
        cout<<"\n1.insert\n2.delete\n3.display\n4.exit\n";
        cin>>ch;
        switch(ch)
        {
            case 1: cout<<"enter the element to add:";
                    cin>>val;
                    q.insert(val);
                    break;
            case 2:q.delet();
                    break;
            case 3:q.display();
                    break;
            case 4:exit(0);
        }
    }
}

```