1. Implement a Stack class for stacks of ints. Include a default constructor, a destructor, and the usual
stack operations: push ( ), pop ( ), is Empty ( ) and is Full( ). Use an array implementation.

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


using namespace std;

class Stack
{
        int data[10];
        int top;
        int Empty()
        {
           if(top==0)
           return 1;

           else
           return 0;
        }


        int Full()
        {
           if(top==9)
           return 1;

           else
           return 0;
        }

    public:

        Stack()
        {
           cout<<"Empty Stack Created";
           top=0;
        }

        ~Stack()
        {
           cout<<"Stack Destroyed";
        }

        void push()
        {
           if(Full())
           {
              cout<<"\nStack is Full";
           }

           else
           {
              cout<<"\nEnter The Integer To Be Pushed : ";
```

```
            cin>>data[top++];
            cout<<"\nPush Successful";
            display();
        }
    }

    void pop()
    {
        if(Empty())
        {
            cout<<"\nPop Failed";
            cout<<"\nStack  is Empty";
        }

        else
        {
            cout<<"\nPop Successful";
            cout<<"\nPopped "<<data[--top];
            display();
        }
    }

    void display()
    {
        if(top==0)
        {
            cout<<"\nStack is Empty";
        }

        else
        {

            cout<<"\n\n\nCurrent Stack : ";
            for(int i=top-1; i>=0; i--)
            {
                cout<<"\n\t"<<data[i];
            }
        }
    }


};


int main()
{

    Stack s;
    int ch;
    do
    {
        cout<<"\n\nEnter Your Choice : ";
        cout<<"\n\t1. Push";
        cout<<"\n\t2. Pop";
        cout<<"\n\t3. Display to Stack";
        cout<<"\n\t4. Exit\n";
```

```
        cin>>ch;

        switch (ch)
        {
            case 1:
                s.push();
                break;
            case 2:
                s.pop();
                break;
            case 3:
                s.display();
                break;
            case 4:
                exit(0);
            default:
                cout<<"\nInvalid Choice";
        }
    }
    while(ch!=4);

    return 0;
}
```

//2. Implement a class, Time. Each object of this class should represent a specic time of day, storing the hours, minutes, and seconds as integers. Write a constructor, access functions, a function to advance the current time of an existing object, a function to reset the current time of an existing object, and a function to display the time of the day.

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

using namespace std;

class Time
{
    int hours;
    int minutes;
    int seconds;

    public:
        Time()
        {
            reset_time();
```

```
    }

    void advance_time()
    {
       int adv_hrs=0, adv_min=0, adv_sec=0;
       cout<<"\n\nAdvance The Time By : -  ";
       cout<<"\nHours : ";
       cin>>adv_hrs;
       cout<<"\nMinutes : ";
       cin>>adv_min;
       cout<<"\nSeconds : ";
       cin>>adv_sec;
       seconds+=adv_sec;
       minutes+=adv_min+seconds/60;
       hours+=adv_hrs+minutes/60;
       seconds%=60;
       minutes%=60;
       hours%=24;
       display();

    }

    void reset_time()
    {

       hours=0;
       minutes=0;
       seconds=0;
       display();

    }

    void display()
    {
       cout<<"\nCurrent Time is : "<<hours<<" : "<<minutes<<" : "<<seconds;
    }

};


int main()
{

   Time t;

int ch;
   do
   {
```

```
        cout<<"\n\nEnter Your Choice : ";
        cout<<"\n\t1. Display The Current Time";
        cout<<"\n\t2. Advance The Time";
        cout<<"\n\t3. Reset The Time";
        cout<<"\n\t4. Exit\n";
        cin>>ch;

        switch (ch)
        {
            case 1:
                t.display();
                break;
            case 2:
                t.advance_time();
                break;
            case 3:
                t.reset_time();
                break;
            case 4:
                exit(0);
            default:
                cout<<"\nInvalid Choice";
        }
    }
    while(ch!=4);


    return 0;
}
```