```
#include <conio.h>
#include <cstdio>
#include <iostream>
#include <string.h>
#include <cstdlib>
using namespace std;
static int p = 0;
class a
{
   char busn[5], driver[10], arrival[5], depart[5], from[10], to[10], seat[8][4][10];
public:
   void install();
   void allotment();
   void empty();
   void show();
   void avail();
   void position(int i);
}
bus[10];
void vline(char ch)
{
   for (int i=80;i>0;i--)
   cout<<ch;
}
void a::install()
{
   cout<<"Enter bus no: ";
   cin>>bus[p].busn;
   cout<<"\nEnter Driver's name: ";
```

```
    cin>>bus[p].driver;
    cout<<"\nArrival time: ";
    cin>>bus[p].arrival;
    cout<<"\nDeparture: ";
    cin>>bus[p].depart;
    cout<<"\nFrom: \t\t\t";
    cin>>bus[p].from;
    cout<<"\nTo: \t\t\t";
    cin>>bus[p].to;
    bus[p].empty();
    p++;
}
void a::allotment()
{
    int seat;
    char number[5];
    top:
    cout<<"Bus no: ";
    cin>>number;
    int n;
    for(n=0;n<=p;n++)
    {
        if(strcmp(bus[n].busn, number)==0)
        break;
    }
    while(n<=p)
    {
        cout<<"\nSeat Number: ";
```

```
        cin>>seat;
        if(seat>32)
        {
           cout<<"\nThere are only 32 seats available in this bus.";
        }
        else
        {
        if (strcmp(bus[n].seat[seat/4][(seat%4)-1], "Empty")==0)
           {
              cout<<"Enter passanger's name: ";
              cin>>bus[n].seat[seat/4][(seat%4)-1];
              break;
           }
        else
           cout<<"The seat no. is already reserved.\n";
           }
           }
        if(n>p)
        {
           cout<<"Enter correct bus no.\n";
           goto top;
        }
     }

void a::empty()
{
    for(int i=0; i<8;i++)
    {
        for(int j=0;j<4;j++)
```

```
        {
            strcpy(bus[p].seat[i][j], "Empty");
        }
    }
}
void a::show()
{
    int n;
    char number[5];
    cout<<"Enter bus no: ";
    cin>>number;
    for(n=0;n<=p;n++)
    {
        if(strcmp(bus[n].busn, number)==0)
        break;
    }
while(n<=p)
{
    vline('*');
    cout<<"Bus no: \t"<<bus[n].busn
    <<"\nDriver: \t"<<bus[n].driver<<"\t\tArrival time: \t"
    <<bus[n].arrival<<"\tDeparture time:"<<bus[n].depart
    <<"\nFrom: \t\t"<<bus[n].from<<"\t\tTo: \t\t"<<
    bus[n].to<<"\n";
    vline('*');
    bus[0].position(n);
    int a=1;
    for (int i=0; i<8; i++)
    {
```

```
        for(int j=0;j<4;j++)
        {
            a++;
            if(strcmp(bus[n].seat[i][j],"Empty")!=0)
            cout<<"\nThe seat no "<<(a-1)<<" is reserved for "<<bus[n].seat[i][j]<<".";
        }
    }
    break;
    }
    if(n>p)
        cout<<"Enter correct bus no: ";
}
void a::position(int l)
{
    int s=0;p=0;
    for (int i =0; i<8;i++)
    {
        cout<<"\n";
        for (int j = 0;j<4; j++)
        {
            s++;
            if(strcmp(bus[l].seat[i][j], "Empty")==0)
                {
                    cout.width(5);
                    cout.fill(' ');
                    cout<<s<<".";
                    cout.width(10);
                    cout.fill(' ');
```

```
                cout<<bus[l].seat[i][j];
                p++;
            }
            else
            {
            cout.width(5);
            cout.fill(' ');
            cout<<s<<".";
            cout.width(10);
            cout.fill(' ');
            cout<<bus[l].seat[i][j];
            }
          }
       }
   cout<<"\n\nThere are "<<p<<" seats empty in Bus No: "<<bus[l].busn;
   }
void a::avail()
{

   for(int n=0;n<p;n++)
   {
      vline('*');
      cout<<"Bus no: \t"<<bus[n].busn<<"\nDriver: \t"<<bus[n].driver
      <<"\t\tArrival time: \t"<<bus[n].arrival<<"\tDeparture Time: \t"
      <<bus[n].depart<<"\nFrom: \t\t"<<bus[n].from<<"\t\tTo: \t\t\t"
      <<bus[n].to<<"\n";
      vline('*');
      vline('_');
   }
```

```
}
int main()
{
system("cls");
int w;
while(1)
{
      //system("cls");
   cout<<"\n\n\n\n\n";
   cout<<"\t\t\t1.Install\n\t\t\t"
   <<"2.Reservation\n\t\t\t"
   <<"3.Show\n\t\t\t"
   <<"4.Buses Available. \n\t\t\t"
   <<"5.Exit";
   cout<<"\n\t\t\tEnter your choice:-> ";
   cin>>w;
   switch(w)
   {
      case 1:   bus[p].install();
         break;
      case 2:   bus[p].allotment();
         break;
      case 3:   bus[0].show();
         break;
      case 4:   bus[0].avail();
         break;
      case 5:   exit(0);
   }
}
```

```
return 0;

}
```