

```

#include<fstream.h>
#include<conio.h>
#include<ctype.h>
#include<string.h>
#include<stdio.h>
#include<dos.h>
#include<stdlib.h>
const char TL=218,TR=191,BL=192,BR=217,HL=196,VL=179;

const char DVL=186,DHL=205,DTL=201,DTR=187,DBL=200,DBR=188;
void EnterDate(char []);
void ClearArea(int C1,int R1,int C2,int R2);
void DispCh(int C,int R,char Ch);
void DispCh(int C,int R,char Ch[]);
void Color(int TC,int TB);
void VLine(int X,int Y,int L);
void HLine(int X,int Y,int L);
void Recta(int C1,int R1,int C2,int R2);
void DVLine(int X,int Y,int L);
void DHLine(int X,int Y,int L);
void DRecta (int C1,int R1,int C2,int R2);
int VMenu(int C,int R,char Men[][15],int Noo);
int HMenu(int C,int R,char Men[][15],int Noo);
//MAIN CLASS

class MyTrain
{
    int Tno, Distance;
    float Fare;
    char TName[30], From[20], To[25], DepartTime[5], ArrivTime[5] ;

public:
    void EnterTrainDetails();
    void ShowTrainDetails();
    int RTno(){return Tno;}
    int RTName(char Name[])
    {
        return strcmpi(TName,Name);
    }
    int RFrom(char F[])
    {
        return strcmpi(From,F);
    }

```

```

    }
    int RTo(char T[])
    {
        return strcmpi(To,T);
    }
    void ReplaceTno()
    {
        DispCh(2,12,"Enter new Train No.");
        cin>>Tno;
    }
    void ReplaceTName()
    {
        DispCh(2,12,"Enter new Train Name ");
        gets(TName);
    }
};

void MyTrain::EnterTrainDetails()
{
    _setcursortype(_NORMALCURSOR);
    DispCh(2,8,"Enter Train No.   : ");
    cin>>Tno;
    DispCh(2,10,"Enter Train Name   : ");
    cin>>TName;
    DispCh(2,12,"Enter Source Station: ");
    gets(From);
    DispCh(2,14,"Enter Destination  : ");
    gets(To);
    DispCh(2,16,"Enter Distance     : ");
    cin>>Distance;
    DispCh(2,18,"Enter the Fare      : ");
    cin>>Fare;
    DispCh(2,20,"Enter Departure Time (xxxx hrs): ");
    cin>>DepartTime;
    DispCh(2,22,"Enter Arrival Time (xxxx hrs): ");
    cin>>ArrivTime;
    _setcursortype(_NOCURSOR);
}

```

```

void MyTrain::ShowTrainDetails()
{
    ClearArea(2,11,78,19);
    DispCh(2,12," Train no.   :");
    gotoxy(30,12);
    cprintf("%d",Tno);
    DispCh(2,13," Train name   :");
    DispCh(30,13,TName);
    DispCh(2,14," Source Station: ");
    DispCh(30,14,From);
    DispCh(2,15," Destination  :");
    DispCh(30,15,To);
    DispCh(2,16," Distance     :");
    gotoxy(30,16);
    cprintf("%d",Distance);
    DispCh(2,17," Fare         :");
    gotoxy(30,17);
    cprintf("%7.2f",Fare);
    DispCh(2,18," Departure time (xxxx hrs): ");
    DispCh(30,18,DepartTime);
    DispCh(2,19," Arrival time (xxxx hrs): ");
    DispCh(30,19,ArrivTime);
}

```

```

void ShowTrains()
{
    MyTrain s;
    fstream fil;
    fil.open("trains.dat", ios::binary | ios::in);
    while(fil.read((char*)&s, sizeof(s)))
    {
        s.ShowTrainDetails();
    }
    fil.close();
}

```

```

void CreateTrains()
{
    MyTrain s;
    char ans;
    fstream fil;

```

```

    fil.open("trains.dat", ios::binary | ios::app);
    do
    {
        s.EnterTrainDetails();
        fil.write((char*)&s, sizeof(s));
        DispCh(2,23,"Enter another record?(Y/N)");
        cin>>ans;
    }
    while(toupper(ans)=='Y');
    fil.close();
}

```

```

void SearchTno()
{
    fstream fil;
    fil.open("trains.dat", ios::binary | ios::in);
    MyTrain T;
    int Tnos,found=0;
    _setcursortype(_NORMALCURSOR);
    DispCh(2,8,"Enter train number to be searched:");
    cin>>Tnos;
    while(fil.read((char*)&T, sizeof(T)) && !found)
    if(Tnos==T.RTno())
    {
        T.ShowTrainDetails();
        found++;
    }
    if(!found)
        DispCh(2,22,"Sorry,no such train number exists ");
    _setcursortype(_NOCURSOR);
    fil.close();
}

```

```

void SearchTName()
{
    fstream fil;
    fil.open("trains.dat", ios::binary | ios::in);
    MyTrain T;
    int found=0;
    char TName[20];

```

```

    _setcursortype(_NORMALCURSOR);
    DispCh(2,8,"Enter train name to be searched:");
    gets(TName);
    while(fil.read((char*)&T, sizeof(T)))
        if(T.RTName(TName)==0)
        {
            T.ShowTrainDetails();
            found++;
        }
    if(!found)
        DispCh(2,22,"Sorry, no such train exists ");
    _setcursortype(_NOCURSOR);
    fil.close();
}

```

```

void SearchFromTo()
{
    fstream fil;
    fil.open("trains.dat", ios::binary | ios::in);
    MyTrain T;
    int found=0;
    char From[20];
    char dest[20];
    _setcursortype(_NORMALCURSOR);
    DispCh(2,8,"Enter From station to be searched:");
    gets(From);
    DispCh(2,9,"Enter Destination to be searched:");
    gets(dest);
    while(!found&&fil.read((char*)&T, sizeof(T)))
        if(T.RFrom(From)==0 && T.RTo(dest)==0)
        {
            T.ShowTrainDetails();
            found++;
        }

    if(!found)
        DispCh(2,22,"Sorry, no such train available ");
    _setcursortype(_NOCURSOR);
    fil.close();
}

```

```

int DeleteTrains()
{
    fstream fil1,fil2;
    fil1.open("trains.dat",ios::binary | ios::in);
    fil2.open("ntrains.dat",ios::binary | ios::out);
    MyTrain t;
    int n,del=0;
    _setcursortype(_NORMALCURSOR);
    DispCh(2,8,"Enter train number to be deleted");
    cin>>n;
    while(fil1.read((char*)&t, sizeof(t)))
    {
        if (t.RTno()!=n)
            fil2.write((char*)&t, sizeof(t));
        else
            del++;
    }
    if(!del)
        DispCh(2,10,"Record not found");
    else
        DispCh(2,10,"Record Deleted");
    _setcursortype(_NOCURSOR);
    fil1.close();
    fil2.close();
    remove("trains.dat");
    rename("ntrains.dat","trains.dat");
    return del;
}

```

```

void ModifyTrainNumber()
{
    MyTrain t;
    fstream fil;
    fil.open("trains.dat",ios::binary | ios::out | ios::in);
    int r,found=0;
    _setcursortype(_NORMALCURSOR);
    ClearArea(2,7,78,23);
    DispCh(2,8,"Enter the train no. to be changed ");
    cin>>r;
    while(!found&&fil.read((char*)&t, sizeof(t)))
    {
        if (r==t.RTno())
        {

```

```

        t.ReplaceTno();
        int byteno=fil.tellp();
        fil.seekp(byteno-sizeof(t));
        fil.write((char*)&t, sizeof(t));
        DispCh(2,10,"The modified record is:");
        t.ShowTrainDetails();
        found++;
    }
}
if(found==0)
    DispCh(2,10,"Invalid Train No.");
_setcursortype(_NOCURSOR);
fil.close();
}

```

```

void ModifyTrainName()
{
    MyTrain t;
    fstream fil;
    fil.open("trains.dat",ios::binary | ios::out | ios::in);
    char r[30];
    int m,found=0;
    ClearArea(2,7,78,23);
    _setcursortype(_NORMALCURSOR);
    DispCh(2,8,"Enter Train no. to be searched");
    cin>>m;
    DispCh(2,9,"Enter the train name to be changed ");
    gets(r);
    while(!found&&fil.read((char*)&t, sizeof(t)))
    {
        if(t.RTName(r)==0 && m==t.RTno())
        {
            t.ReplaceTName();
            int byteno=fil.tellp();
            fil.seekp(byteno-sizeof(t));
            fil.write((char*)&t, sizeof(t));
            DispCh(2,10,"The modified record is:");
            t.ShowTrainDetails();
            found++;
        }
    }
}

```

```

        if(found==0)
            DispCh(2,10,"Invalid Train Name");
        _setcursortype(_NOCURSOR);
        fil.close();
    }

```

```

class Passenger
{
    char PName[30], Gender, Father[30], FromStation[30], ToStation[30], Date[10];
    int Age, Tno;
    int SeatNo;
public:
    void InputPassengerDetails();
    void OutputPassengerDetails();
    void MatchToFrom(int);
    void AssignSeatNo(int n)
    {
        SeatNo=n;
    }
    int RSeatNo()
    {
        return SeatNo;
    }

    int RTno()
    {
        return Tno;
    }

    int RTName(char TName[])
    {
        return strcmpi(PName,TName);
    }
};

```

```

void EnterDate(char D[])
{
    D[0] = getche();
    D[1] = getche();
    cprintf("/");
}

```



```

        D[2]=getche();
        D[3]=getche();
        cprintf("/");
        D[4]=getche();
        D[5]=getche();
        D[6] ='\0';
        getch();
    }

```

```

void Passenger::InputPassengerDetails()
{

```

```

    _setcursortype(_NORMALCURSOR);
    ClearArea(2,8,78,23);
    DispCh(30,8,"ENTER PASSENGER DETAILS");
    DispCh(2,10,"Enter Name: ");
    gets(PName);
    DispCh(2,11,"Enter Age:");
    cin>>Age;
    DispCh(2,12,"Enter Gender(M/F): ");
    cin>>Gender;
    DispCh(2,13,"Enter father's Name: ");
    gets(Father);
    DispCh(2,14,"Enter Date of travel:eg.(22/10/08) ");
    EnterDate(Date);
    DispCh(2,15,"Enter Station From : ");
    gets(FromStation);
    DispCh(2,16,"Enter Destination Station: ");
    gets(ToStation);
    DispCh(2,17,"Enter Train no. ");
    cin>>Tno;
}

```

```

void Passenger::OutputPassengerDetails()
{

```

```

    ClearArea(2,10,78,23);
    DispCh(2,10,"Name  :");
    DispCh(30,10,PName);
    DispCh(2,11,"Age  :");
    gotoxy(30,11);
    cprintf("%d",Age);
    DispCh(2,12,"Gender :");

```

```

DispCh(30,12,Gender);
DispCh(2,13,"Father's Name : ");
DispCh(30,13,Father);
DispCh(2,14,"Date of Travel : ");
DispCh(30,14,Date[0]);
DispCh(31,14,Date[1]);
DispCh(32,14,'/');
DispCh(33,14,Date[2]);
DispCh(34,14,Date[3]);
DispCh(35,14,'/');
DispCh(36,14,Date[4]);
DispCh(37,14,Date[5]);
DispCh(2,15,"Source Station: ");
DispCh(30,15,FromStation);
DispCh(2,16,"Destination Station: ");
DispCh(30,16,ToStation);
DispCh(2,17,"Train no. ");
gotoxy(30,17);
cprintf("%d",Tno);
}

```

```

void Reservation(int t,char d[])
{
    char name[20];
    char TN[10];
    itoa(t,TN,10);
    strcpy(name,TN);
    strcat(name,d);
    strcat(name,".DAT");
    fstream fil;
    fil.open(name,ios::binary|ios::app);
    Passenger k;
    int n;
    fil.seekg(0,ios::end);
    int byteno=fil.tellg()/(sizeof(k));
    n=byteno+1;
    k.AssignSeatNo(n);
    k.InputPassengerDetails();
    fil.write((char*)&k,sizeof(k));
    DispCh(2,20,"Your seat number is ");
    gotoxy(30,20);
    cprintf("%d",n);
    fil.close();
}

```

```
}
```

```
void Cancellation(int t,char d[])
```

```
{
```

```
    char name[20];
    char TN[10];
    itoa(t,TN,10);
    strcpy(name,TN);
    strcat(name,d);
    strcat(name,".DAT");
    fstream fil;
    fstream fil1,fil2;
    fil1.open(name,ios::binary | ios::in);
    fil2.open("nname.dat",ios::binary | ios::out);
    Passenger z;
    int n,del=0;
    DispCh(2,18,"Enter Your Seat No.");
    cin>>n;
    while(fil1.read((char*)&z, sizeof(z)))
    {
        if (z.RSeatNo()!=n)
            fil2.write((char*)&z, sizeof(z));
        else
            del++;
    }
    if(!del)
        DispCh(2,22,"Invalid seat Number!!");
    else
        DispCh(2,22,"Your ticket has been cancelled!");

    fil1.close();
    fil2.close();

    remove(name);
    rename("nname.dat", name);
```

```
}
```

```
void Passenger::MatchToFrom(int c)
```

```
{
```

```
    fstream fil;
    fil.open("trains.dat", ios::binary | ios::in);
    MyTrain t;
    Passenger s;
```

```

char ss[10], ds[10];
int found=0;
_setcursortype(_NORMALCURSOR);
DispCh(2,8,"Please enter the following information");
DispCh(2,10,"Source station");
gets(ss);
DispCh(2,12,"Destination station");
gets(ds);
while(!found&&fil.read((char*)&t, sizeof(t)))
{
    if(t.RFrom(ss)==0 && t.RTo(ds)==0)
    {
        DispCh(2,14,"Your train no. is ");
        gotoxy(30,14);
        cprintf("%d",t.RTno());
        DispCh(2,16,"Enter Date of travel:eg.(22/10/08) ");
        EnterDate(Date);
        if(c==0)
            Reservation(t.RTno(),Date);
        else
            if(c==1)
                Cancellation(t.RTno(),Date);
        found++;
    }
}

if(found==0)
    DispCh(2,14,"No Train Available");
_setcursortype(_NOCURSOR);

fil.close();
}

```

```

void SearchPassengerName()
{
    fstream fil;

    Passenger T;
    int found=0;
    char pname[20];
    _setcursortype(_NORMALCURSOR);
    ClearArea(2,7,78,23);
    DispCh(2,8,"Enter Passenger Name to be searched:");
}

```

```

    gets(pname);
    DispCh(2,9,"Enter Train Number in which travelling:");
    char Tn[20];
    gets(Tn);
    DispCh(2,10,"Enter Date on which travelling:");
    char Dt[20];
    EnterDate(Dt);
    char Fname[20];
    strcpy(Fname,Tn);
    strcat(Fname,Dt);
    strcat(Fname,".DAT");
    fil.open(Fname, ios::binary | ios::in);
    while(fil.read((char*)&T, sizeof(T)))
        if(T.RTName(pname)==0)
        {
            T.OutputPassengerDetails();
            found++;
        }
    if(!found)
        DispCh(2,22,"Sorry, no such passenger exists ");
    _setcursortype(_NOCURS);

fil.close();
}

void SearchPassengerTrainSeat()
{
    fstream fil;
    Passenger T;
    int tnos,snos,found=0;
    _setcursortype(_NORMALCURSOR);
    DispCh(2,9,"Enter Train Number in which travelling:");
    int Tnos;
    char Tn[20];
    cin>>Tnos;
    itoa(Tnos,Tn,10);
    DispCh(2,10,"Enter Date on which travelling:");
    char Dt[20];
    EnterDate(Dt);
    char Fname[20];
    strcpy(Fname,Tn);
    strcat(Fname,Dt);
    strcat(Fname,".DAT");

```

```

fil.open(Fname, ios::binary | ios::in);

ClearArea(2,7,78,23);
DispCh(2,9,"Enter seat number to be searched:");
cin>>snos;
while(!found&&fil.read((char*)&T, sizeof(T)))
if(Tnos==T.RTno()&&snos==T.RSeatNo())
{
    T.OutputPassengerDetails();
    found++;
}
if(!found)
    DispCh(2,22,"Sorry,no such allocation exists ");
fil.close();
}

```

```

void main()
{
    _setcursortype(_NOCURSORS);
    textmode(C80);
    Color(WHITE,CYAN);
    ClearArea(1,1,80,25);
    DRecta(1,1,80,24);
    DispCh(12,2,"RAILWAY RESERVATION SYSTEM");
    HLine(2,3,76);
    char Menu[][15]={    "ADMIN ",
                        "SEARCH ",
                        "UTILITY",
                        "QUIT  "
                        };

    char SMenu[][15]={  "ADD TRAINS ",
                        "DELETE TRAINS",
                        "VIEW TRAINS ",
                        "MODIFY TRAINS"
                        };

    char LMenu[][15]={  "BY TRAIN NO ",
                        "BY TRAIN NAME",
                        "BY SOURCE STN",
                        "BY PSNGR NAME",
                        "BY SEAT NO. "

```

```

};

char MMenu[][15]={ "RESERVATION ",
                   "CANCELLATION "

};

char NMenu[][15]={ "BY NUMBER ",
                   "BY NAME  "

};

int Quit=0;
do
{
    Color(BLUE,WHITE);
    ClearArea(2,4,79,23);
    int N=HMenu(2,4,Menu,4),SN;
    DispCh(30,23,"Press Esc To Go Back...");
    switch (N)
    {
        case 0:
            SN=VMenu(2,6,SMenu,4);
            switch(SN)
            {
                case 0: ClearArea(2,4,79,23);
                        DispCh(30,6,"ENTER TRAIN INFORMATION");
                        CreateTrains();
                        break;

                case 1: ClearArea(2,4,79,23);
                        DispCh(30,6,"DELETING TRAIN");
                        int del;
                        del=DeleteTrains();
                        if(del!=0)
                        {
                            rename("ntrains.dat", "trains.dat");
                            remove("ntrains.dat");
                        }
                        break;

                case 2: ClearArea(2,4,79,23);
                        DispCh(30,6,"VIEW TRAINS");

```

```

        ShowTrains();
        break;

    case 3: ClearArea(2,4,79,23);
        DispCh(30,6,"MODIFY TRAINS");
        int X= VMenu(2,7,NMenu,2);
        switch(X)
        {
            case 0:ModifyTrainNumber();
                break;
            case 1:ModifyTrainName();
        }
        break;

    }
    getch();
    break;

case 1:
    SN=VMenu(25,6,LMenu,5);
    switch(SN)
    {
        case 0: ClearArea(2,4,79,23);
            DispCh(30,6,"SEARCHING BY TNO");
            SearchTno();
            break;
        case 1: ClearArea(2,4,79,23);
            DispCh(30,6,"SEARCHING BY TNAME");
            SearchTName();
            break;
        case 2: ClearArea(2,4,79,23);
            DispCh(30,6,"SEARCHING BY STN");
            SearchFromTo();
            break;
        case 3: ClearArea(2,4,79,23);
            DispCh(30,6,"SEARCHING BY NAME");
            SearchPassengerName();
            break;
        case 4: ClearArea(2,4,79,23);
            DispCh(30,6,"SEARCHING BY SEAT");
            SearchPassengerTrainSeat();
            break;
    }

```



```

        getch();
        break;

    case 2:
        SN=VMenu(50,6,MMenu,2);
        Passenger s;
        switch(SN)
        {
            case 0: ClearArea(2,4,79,23);
                    DispCh(30,6,"BOOKING OF TICKET");
                    break;
            case 1: ClearArea(2,4,79,23);
                    DispCh(30,6,"CANCELLATION");

        }
        s.MatchToFrom(SN);
        getch();
        break;

    case 3:
    case -1:
        Quit=1;
    }
}
while (!Quit);
getch();
}

void DispCh(int C,int R,char Ch)
{
    gotoxy(C,R);cprintf("%c",Ch);//C 1-80 R 1-25
}

void DispCh(int C,int R,char Ch[])
{
    gotoxy(C,R);cprintf("%s",Ch);
}

void Color(int TC,int TB)
{
    textcolor(TC);
    textbackground(TB);
}

```

```

void VLine(int X,int Y,int L)
{
    for (int i=Y;i<=Y+L;i++)
        DispCh(X,i,VL);
}

```

```

void HLine(int X,int Y,int L)
{
    for (int i=X;i<=X+L;i++)
        DispCh(i,Y,HL);
}

```

```

void Recta (int C1,int R1,int C2,int R2)
{
    DispCh(C1,R1,TL);
    DispCh(C1,R2,BL);
    DispCh(C2,R1,TR);
    DispCh(C2,R2,BR);
    VLine(C1,R1+1,R2-R1-2);
    VLine(C2,R1+1,R2-R1-2);
    HLine(C1+1,R1,C2-C1-2);
    HLine(C1+1,R2,C2-C1-2);
}

```

```

void DVLine(int X,int Y,int L)
{
    for (int i=Y;i<=Y+L;i++)
        DispCh(X,i,DVL);
}

```

```

void DHLine(int X,int Y,int L)
{
    for (int i=X;i<=X+L;i++)
        DispCh(i,Y,DHL);
}

```

```

void DRecta (int C1,int R1,int C2,int R2)
{
    DispCh(C1,R1,DTL);
    DispCh(C1,R2,DBL);
    DispCh(C2,R1,DTR);
}

```

```

DispCh(C2,R2,DBR);
DVLine(C1,R1+1,R2-R1-2);
DVLine(C2,R1+1,R2-R1-2);
DHLine(C1+1,R1,C2-C1-2);
DHLine(C1+1,R2,C2-C1-2);
}

```

```

int VMenu(int C,int R,char Men[][15],int Noo)
{

```

```

    Color(WHITE,BLUE);
    Recta(C,R,C+strlen(Men[0])+1,R+Noo+1);
    for (int i=0;i<Noo;i++)
        DispCh(C+1,R+i+1,Men[i]);

```

```

    int Exit=0,Sel=0;

```

```

    do
    {

```

```

        int Tsel=Sel;
        Color(BLUE,WHITE);
        DispCh(C+1,R+Sel+1,Men[Sel]);
        Color(WHITE,BLUE);
        char Ch=getch();
        switch(Ch)
        {

```

```

            case 71:

```

```

                Sel=0;break;    //Home -> Move to first option

```

```

            case 79:

```

```

                Sel=Noo-1;break; //End -> Move to Last option

```

```

            case 72:

```

```

                if (Sel==0)    //Up -> Checking if already on top

```

```

                    Sel=Noo-1; //    Move to the last option

```

```

                else

```

```

                    Sel--;    //    Move one option up

```

```

                break;

```

```

            case 80:

```

```

                if (Sel==Noo-1) //Down -> Checking if already at bottom

```

```

                    Sel=0;    //    Move to the first option

```

```

                else

```

```

                    Sel++;    //    Move one step down

```

```

                break;

```

```

            case 27:

```

```

                Sel=-1;    //Escape-> Getting out of the Menu without valid

```

selection

```
                case 13:
                    Exit=1;    //Enter-> Getting out of the Menu with current selection
                }
                DispCh(C+1,R+Tsel+1,Men[Tsel]);
            }
            while (!Exit);
            return Sel;        //Returning the selected value from menu
        }
```

int HMenu(int C,int R,char Men[][15],int Noo)

```
{
    int Exit=0,Sel=0;
    Color(BLUE,WHITE);
    for (int i=0;i<Noo;i++)
        DispCh(C+21*i+1,R,Men[i]);
    do
    {
        int Tsel=Sel;
        Color(WHITE,BLUE);
        DispCh(C+21*Sel+1,R,Men[Sel]);
        Color(BLUE,WHITE);
        char Ch=getch();
        switch(Ch)
        {
            case 71:
                Sel=0;break;    //Home
            case 79:
                Sel=Noo-1;break; //End
            case 75:
                if (Sel==0)    //Left
                    Sel=Noo-1;
                else
                    Sel--;
                break;
            case 77:
                if (Sel==Noo-1) //Right
                    Sel=0;
                else
                    Sel++;
                break;
            case 27:
                Sel=-1; //Escape
        }
    }
    while (Exit==0);
    return Sel;
}
```

```

                case 13:
                    Exit=1; //Enter
                }
            DispCh(C+21*Tsel+1,R,Men[Tsel]);
        }
        while (!Exit);
        return Sel;
    }

void ClearArea(int C1,int R1,int C2,int R2)
{
    window(C1,R1,C2,R2);
    clrscr();
    window(1,1,80,25);
}

//Enter-> Getting out of the Menu with current selection
    }
    DispCh(C+1,R+Tsel+1,Men[Tsel]);
}
while (!Exit);
return Sel;          //Returning the selected value from menu
}

int HMenu(int C,int R,char Men[][15],int Noo)
{
    int Exit=0,Sel=0;
    Color(BLUE,WHITE);
    for (int i=0;i<Noo;i++)
        DispCh(C+21*i+1,R,Men[i]);
    do
    {
        int Tsel=Sel;
        Color(WHITE,BLUE);
        DispCh(C+21*Sel+1,R,Men[Sel]);
        Color(BLUE,WHITE);
        char Ch=getch();
        switch(Ch)
        {
            case 71:
                Sel=0;break;

```