

**C++ PROJECT**

**RAILWAY**

**RESERVATION**

**SYSTEM**

**MADE BY – ANANT MATHUR**

**SUBMITTED BY – ANANT MATHUR  
(XII-A1)**

# INDEX

<b>S. NO.</b>	<b>CONTENT</b>	<b>PAGE NUMBER</b>
<b>1</b>	CERTIFICATION	2
<b>2</b>	ACKNOWLEDGEMENT	3
<b>3</b>	INTRODUCTION TO THE PROJECT	4
<b>4</b>	SYSTEM REQUIREMENTS	5
<b>5</b>	DEVELOPMENT LANGUAGE USED	6
<b>6</b>	HEADER FILES USED	8
<b>7</b>	CLASSES	9
<b>8</b>	MENU MAP	10
<b>9</b>	FLOW CHARTS	11
<b>10</b>	CODING	25
<b>11</b>	OUTPUTS	55
<b>12</b>	BIBLIOGRAPHY	73

# **CERTIFICATION**

This is to certify that the project work  
**“RAILWAY RESERVATION SYSTEM”**  
is a bonafide record of work done by

**ANANT MATHUR (XII-A1)**

under my guidance and supervision.

***MR. ABHAY KUMAR***

***PGT, COMPUTER SCIENCE***

***LAXMAN PUBLIC SCHOOL***

# **ACKNOWLEDGMENT**

I am extremely grateful to **Mr. ABHAY KUMAR**, Teacher of Department of Computer Science for his able guidance and useful suggestions, which helped me in completing the project work, in time.

I would also like to thank all the staff members of Computer Science department who helped me directly or indirectly in the completion of this project .

Finally, yet importantly, I would like to express my heartfelt thanks to my beloved parents for their blessings, my friends/classmates for their help and wishes for the successful completion of this project.

**ANANT MATHUR**

# **INTRODUCTION**

## **“RAILWAY RESERVATION SYSTEM”**

The project is designed for Railway System in C++. The title of the project is **Railway Reservation system.** In this project a person can check the availability of a train and book tickets by entering the boarding point and destination.

A ticket will be printed containing the PNR number. The PNR Number can also be used for cancellation of the ticket.

Administrator of the project, with the help of a password, can enter new train record, display all train records, modify train records and delete train records.

# **REQUIREMENTS**

## **Software Requirement:-**

<b><u>Software</u></b>	<b><u>Description</u></b>
Windows	Operating System
Turbo C++ 4.5	For execution of the program
Ms Word	For presentation of the output

## **Hardware Requirement:-**

<b><u>Hardware</u></b>	<b><u>Description</u></b>
Ram	256 MB
Hard Disk	20 GB
CD ROM	400 MB
Processor	Pentium III
Monitor	14.4"
Keyboard	104 keys

# **DEVELOPMENT** **LANGUAGE** **USED**

*I opted C++ as the development language for my project because C++ is a versatile language for handling very large programs. C++ follows all the standards given by OOPs.*

## **Advantages of using C++ in the project:-**

Using C++ (an Object Oriented Programming language) allowed breaking complex large software programs to simpler, smaller and manageable components.

The following are some advantages of using OOPs:-

**MODULAR DESIGN:-**The software developed for the Railway Reservation System around OOP is modular, because this is built on objects and we know objects are entity in themselves, whose internal working is hidden from other objects.

**MODIFIABLE:-**Because of its inherent properties of data abstraction and encapsulation the internal working of objects is hidden from other objects in the program. So any modification made in an object should not affect the rest of the system.

**EXTENSIBLE:-**If further enhancement in the existing Railway Reservation System is required for the its adaption in a new environment then it can easily be done by simply adding new features in old class types.



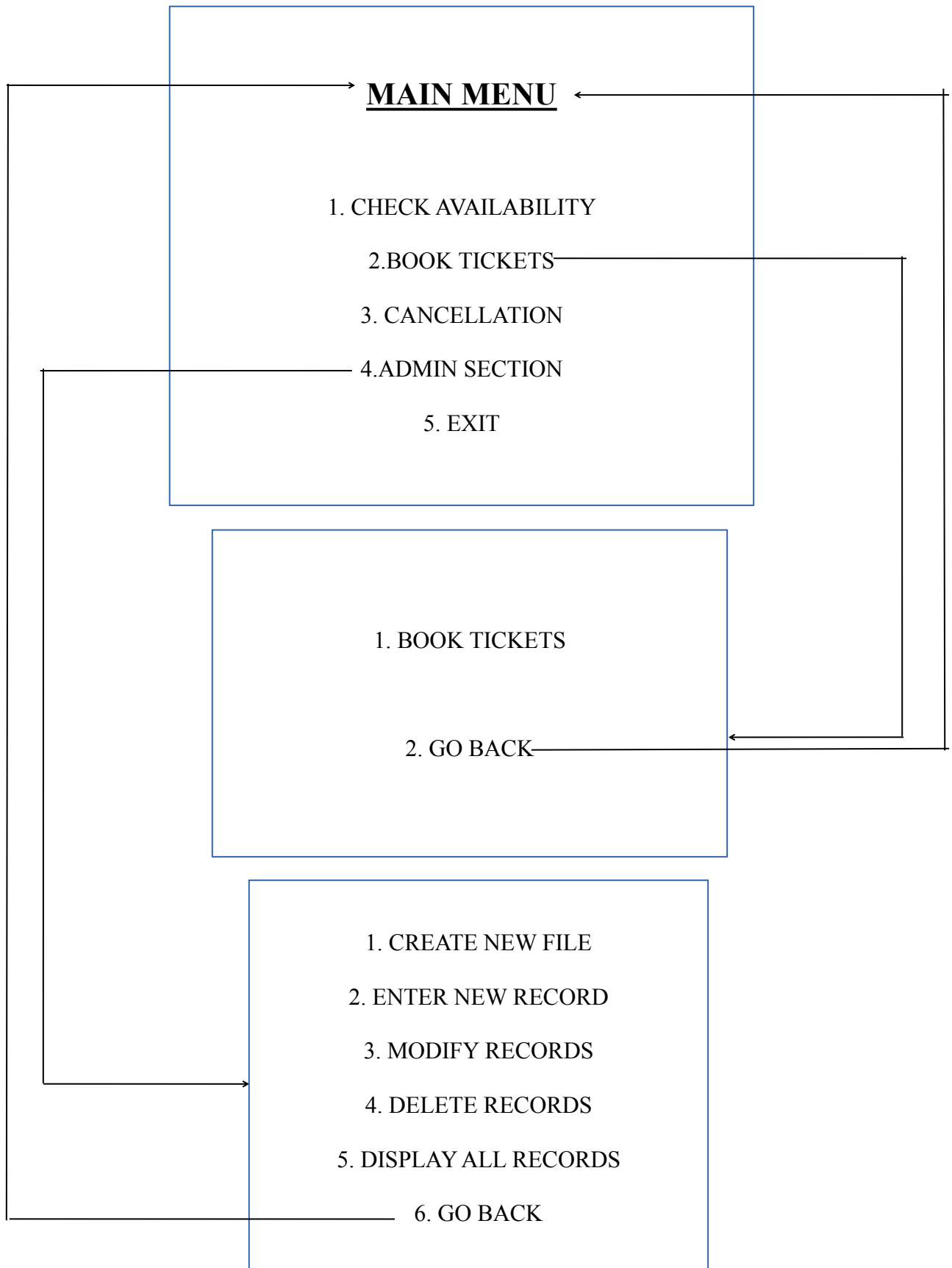
# HEADER FILES USED

- i.) `#include<fstream.h>` - For basic input, output and file handling functions
- ii.) `#include<conio.h>` - For `getch()` and `clrscr()` functions
- iii.) `#include<stdio.h>` - For `gets()` function
- iv.) `#include<string.h>` - For string functions such as `strcmpi()`
- v.) `#include<process.h>` - For `exit(0)` function

# CLASSES

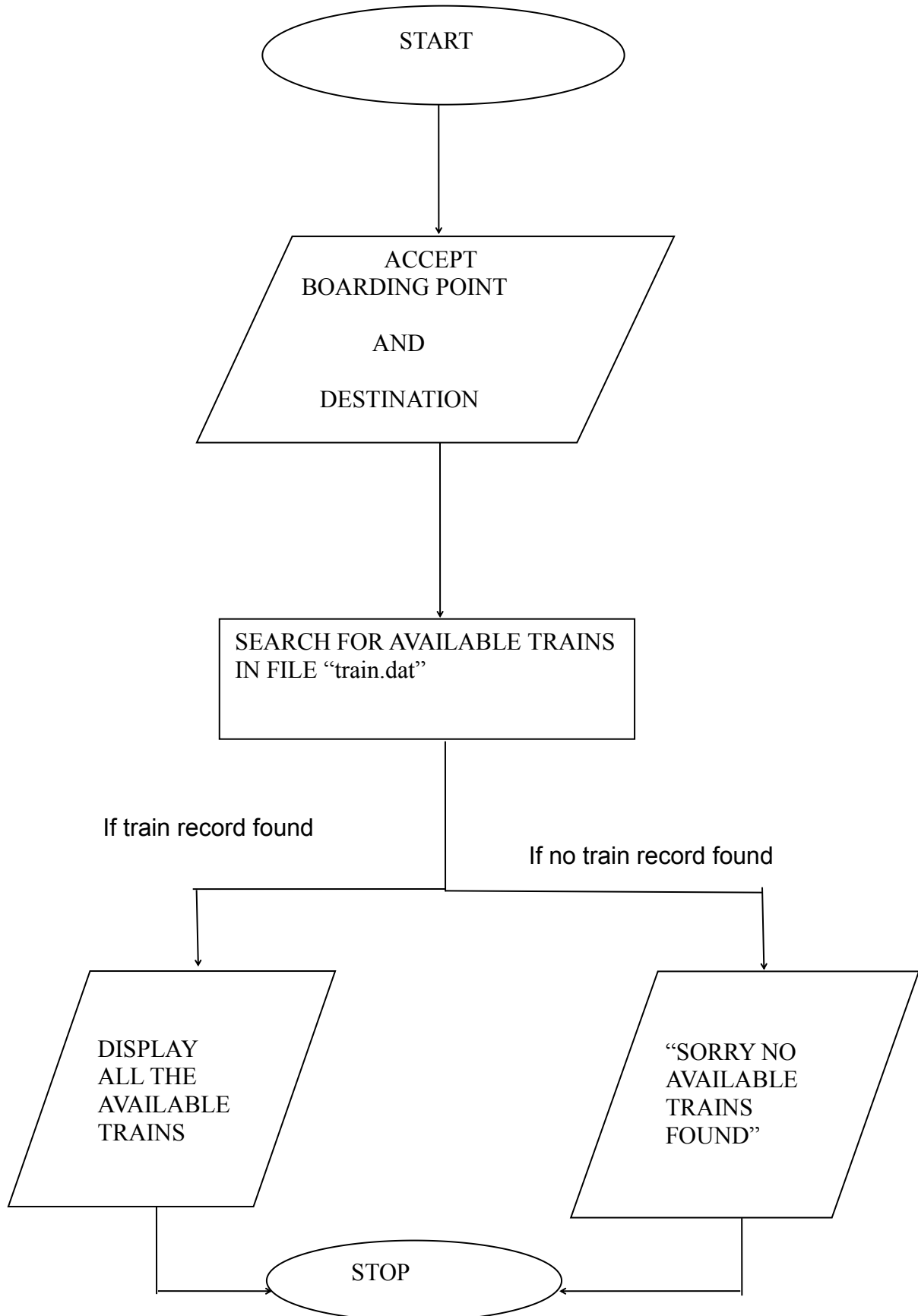
- i.) **pn** – For PNR (passenger name record) number
  
- ii.) **train** – To store, display, delete and modify train records
  
- iii.) **book** – To accept, calculate, show and delete booking details
  
- iv.) **passenger** – To accept and show passenger details

# MENU MAP

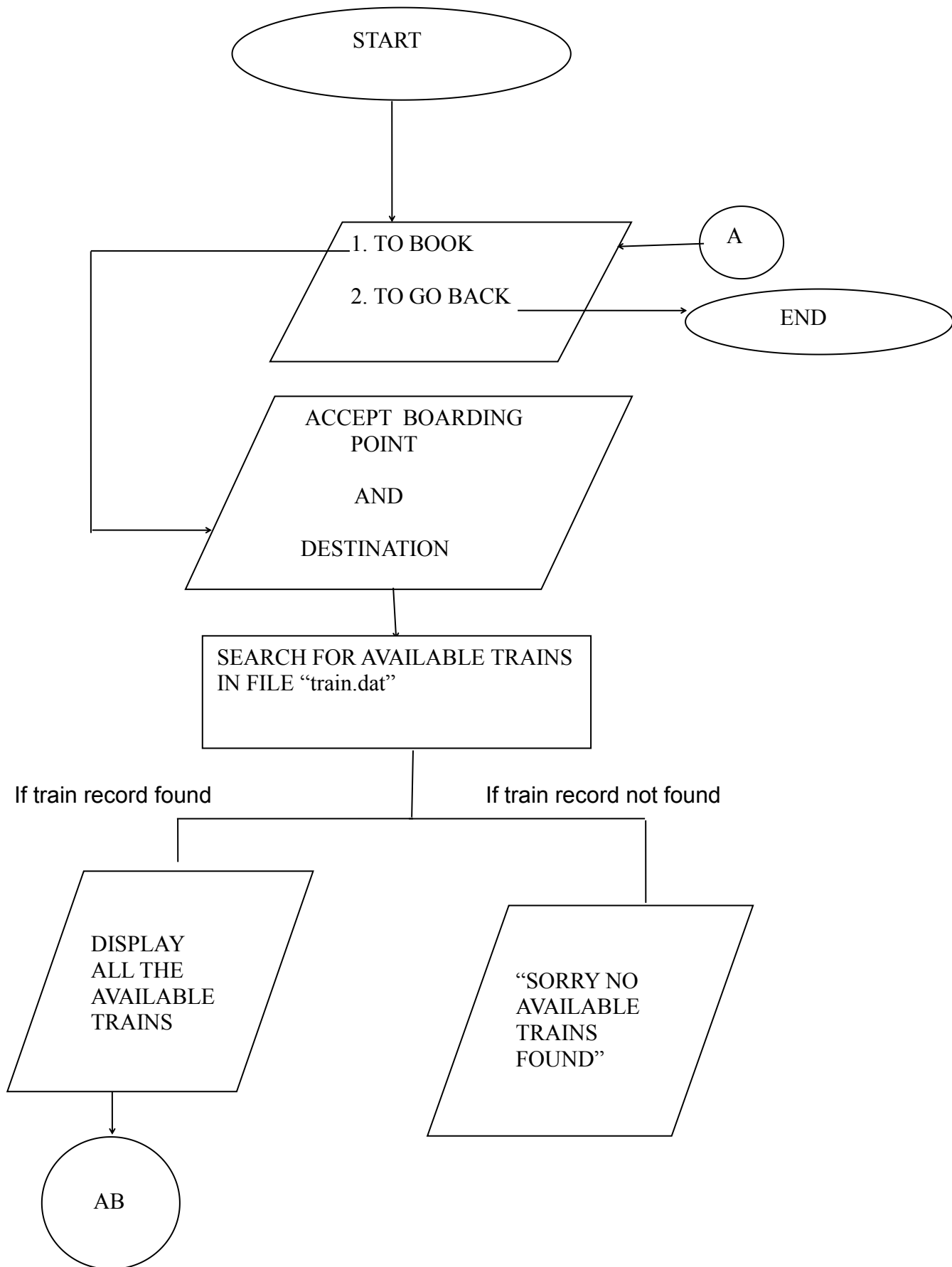


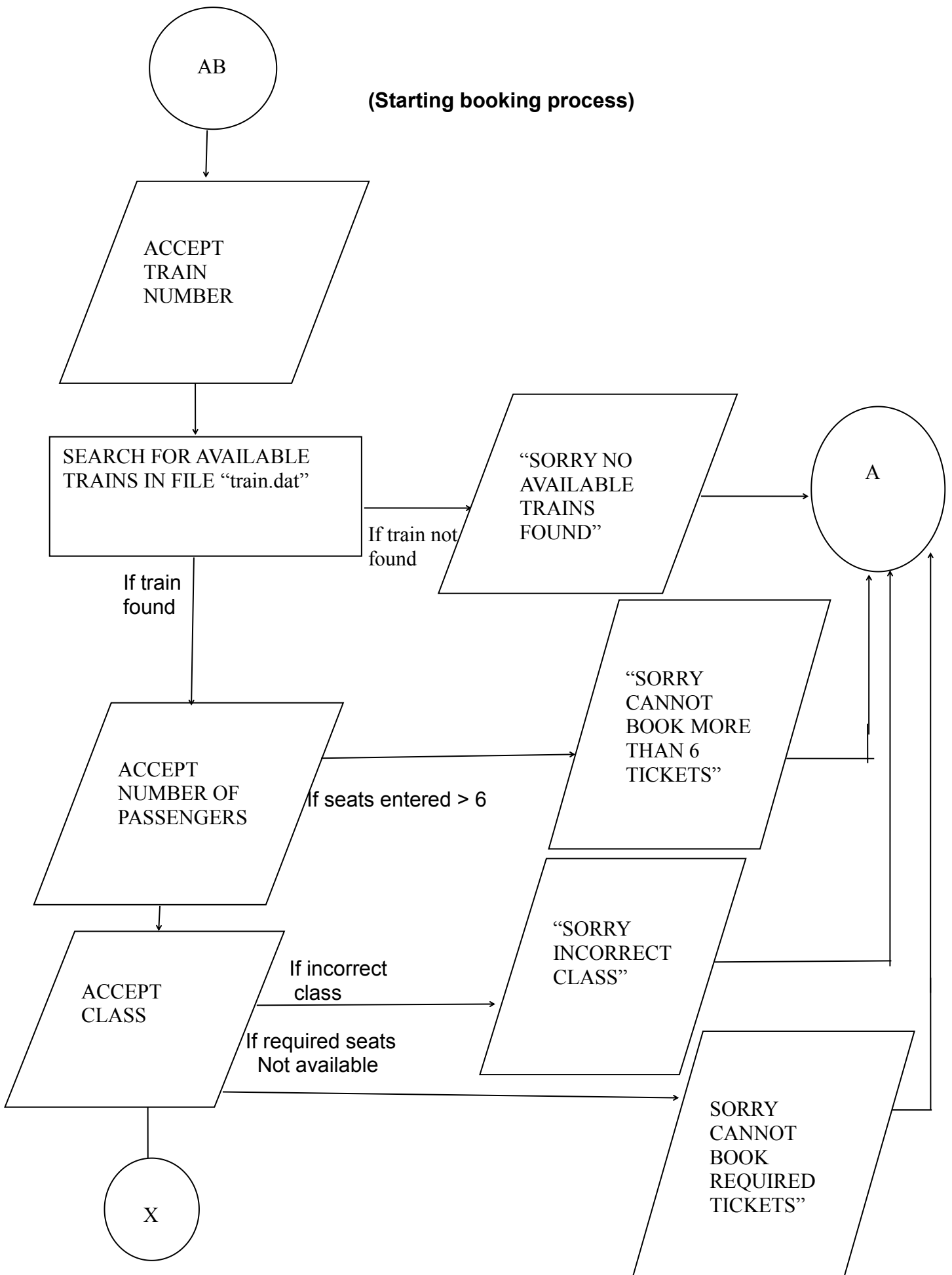
# FLOW CHARTS

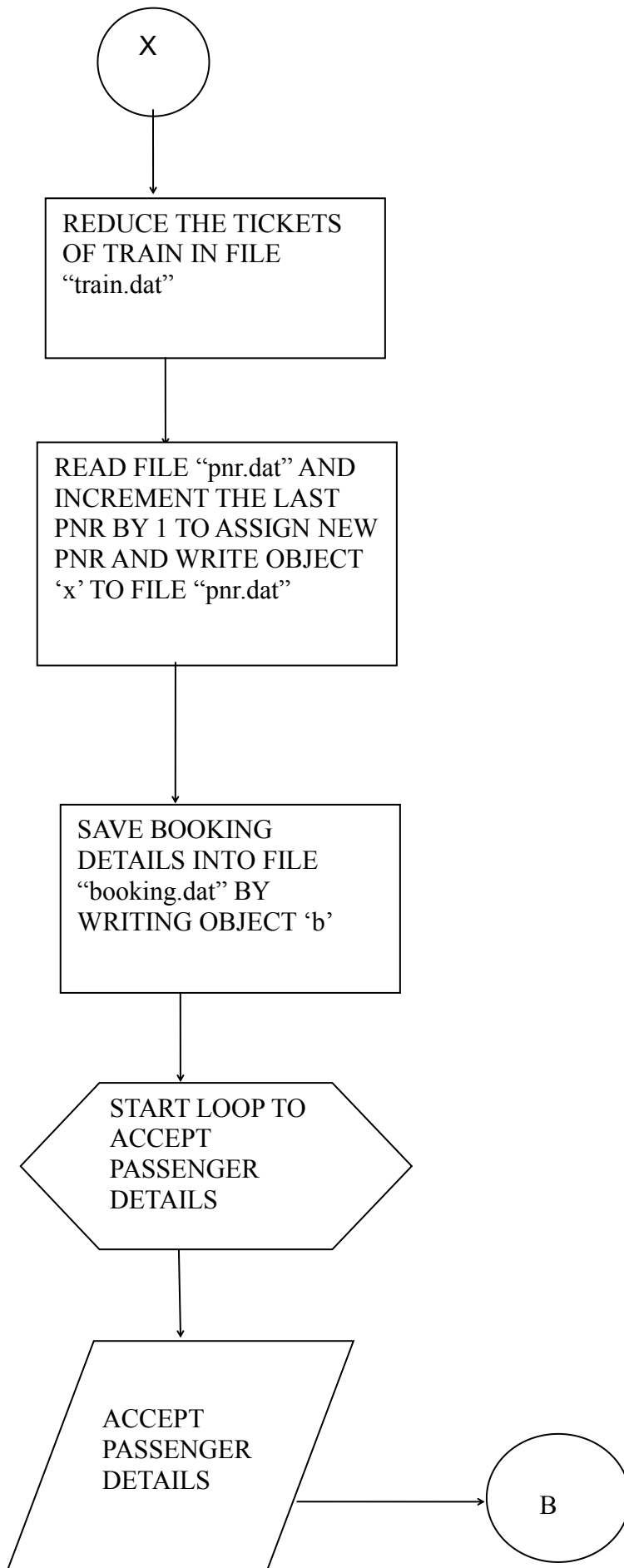
## 1. TO CHECK TICKET AVAILABILITY

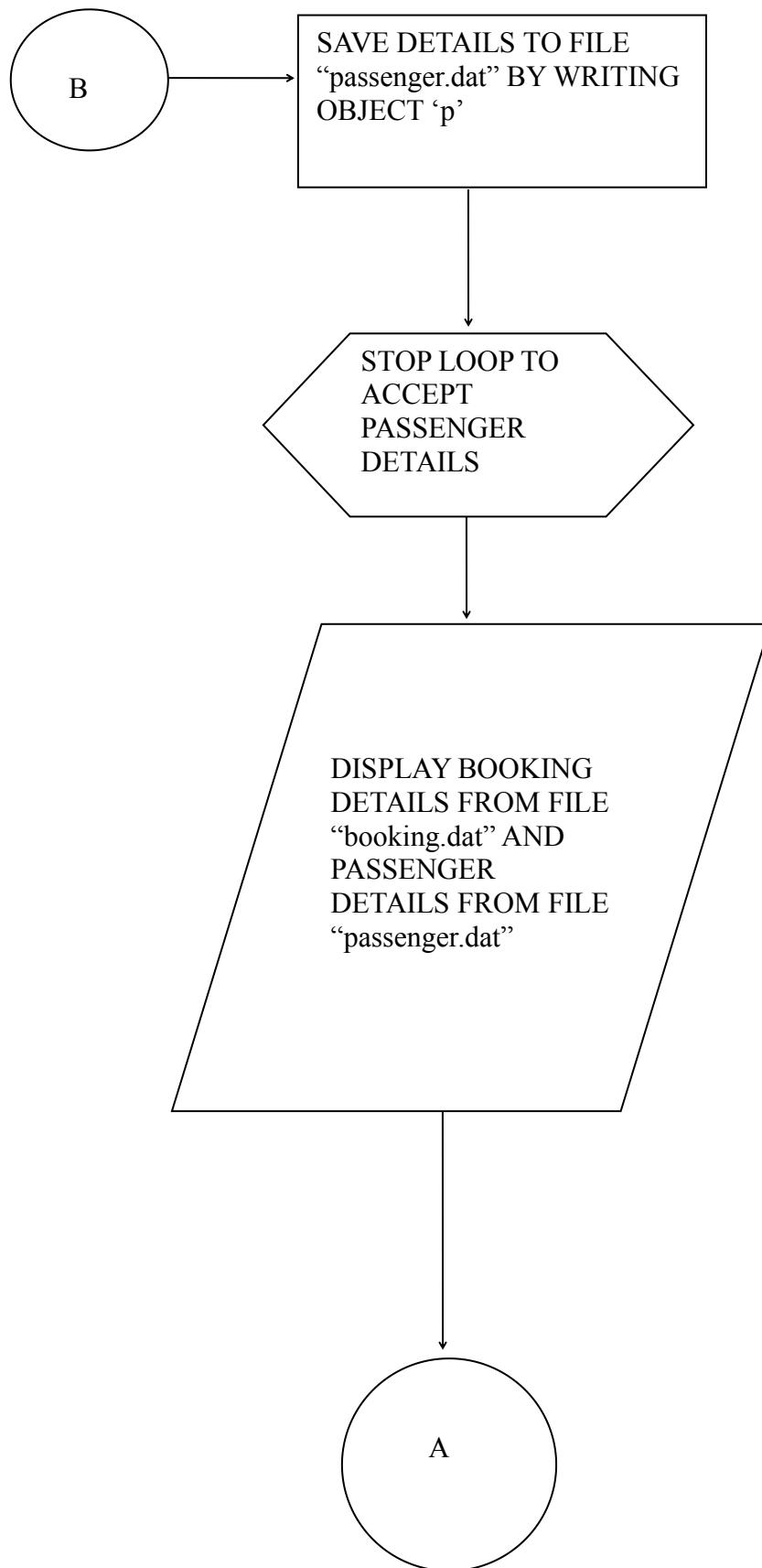


## 2. BOOK TICKETS



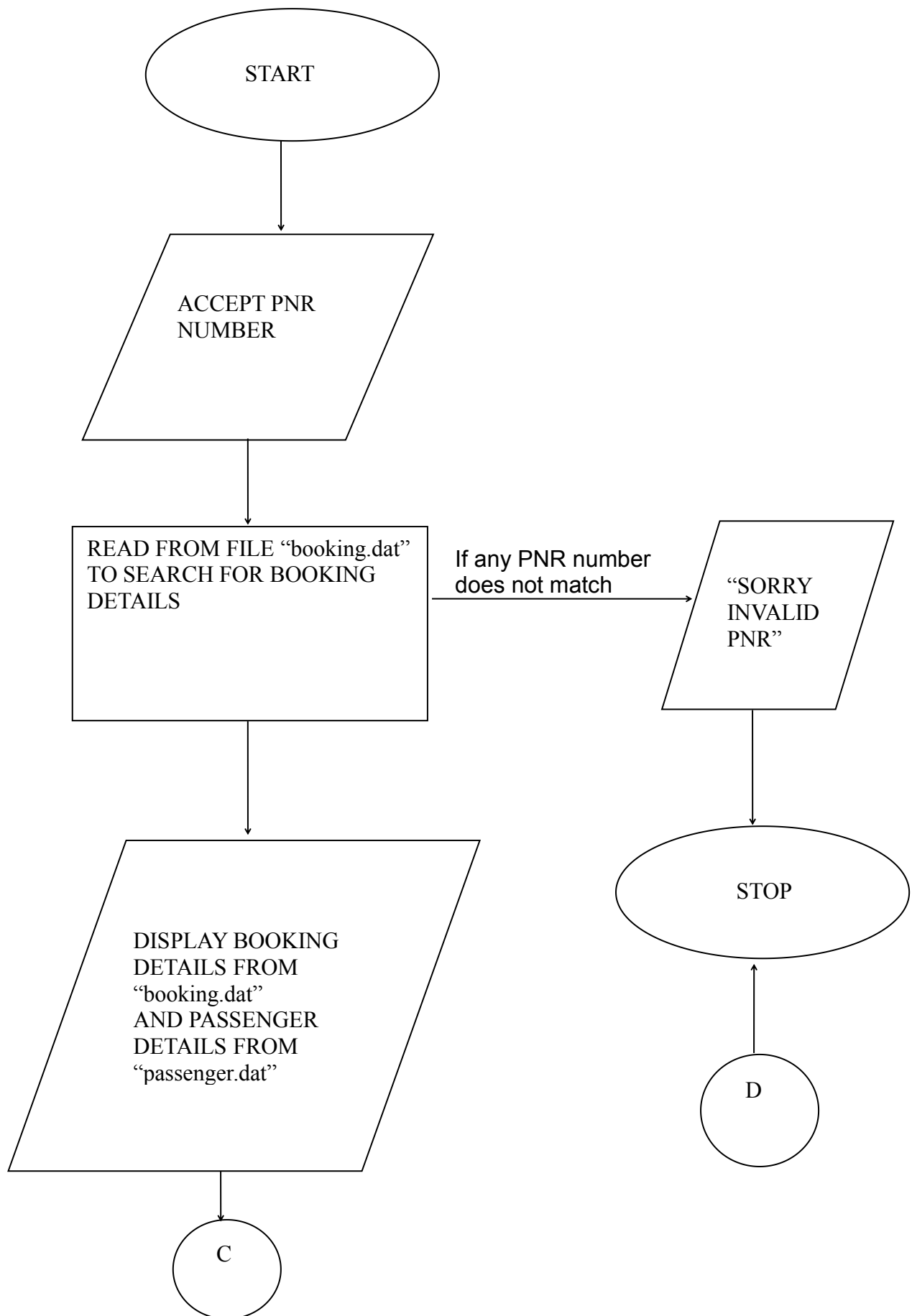


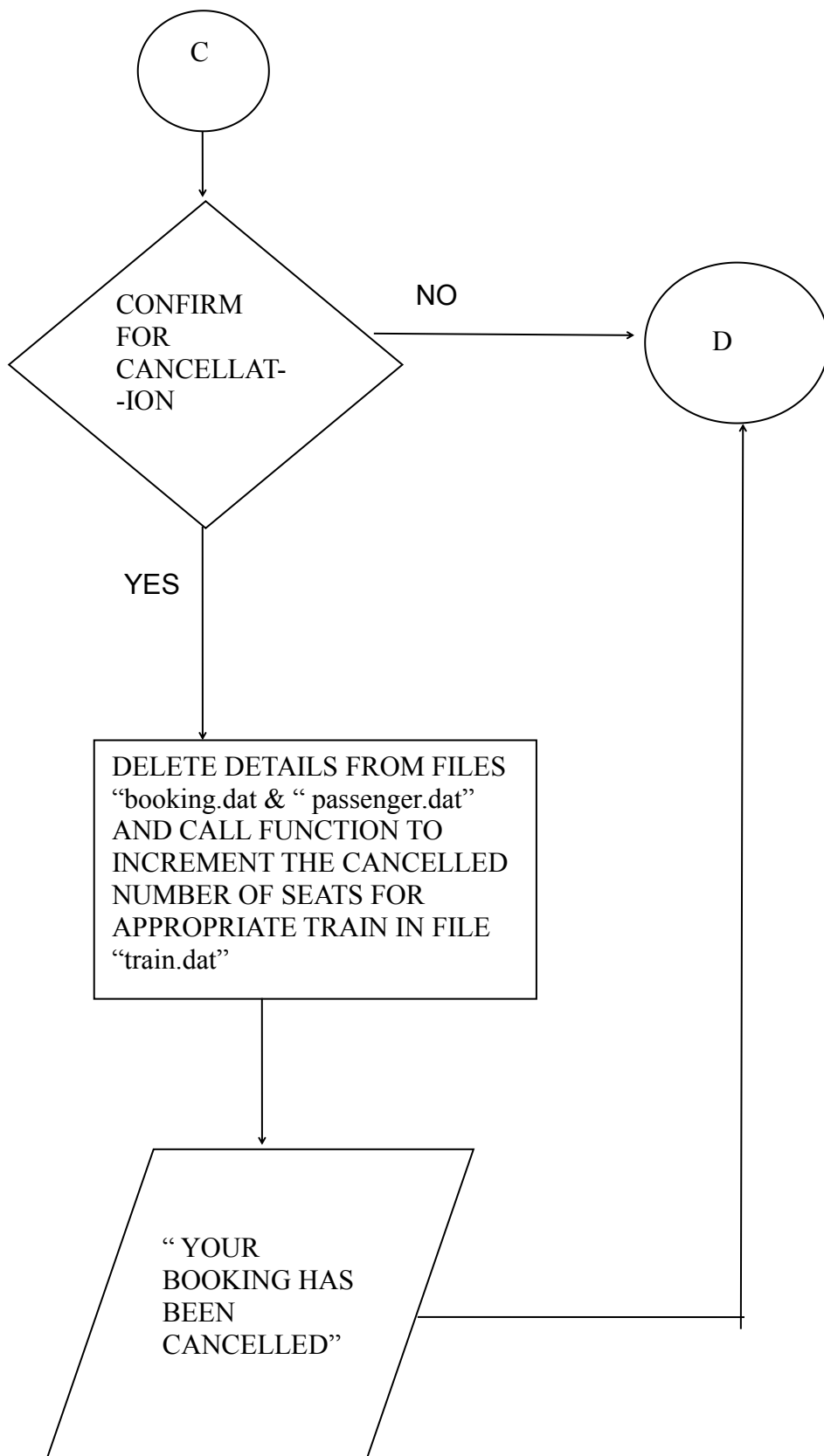




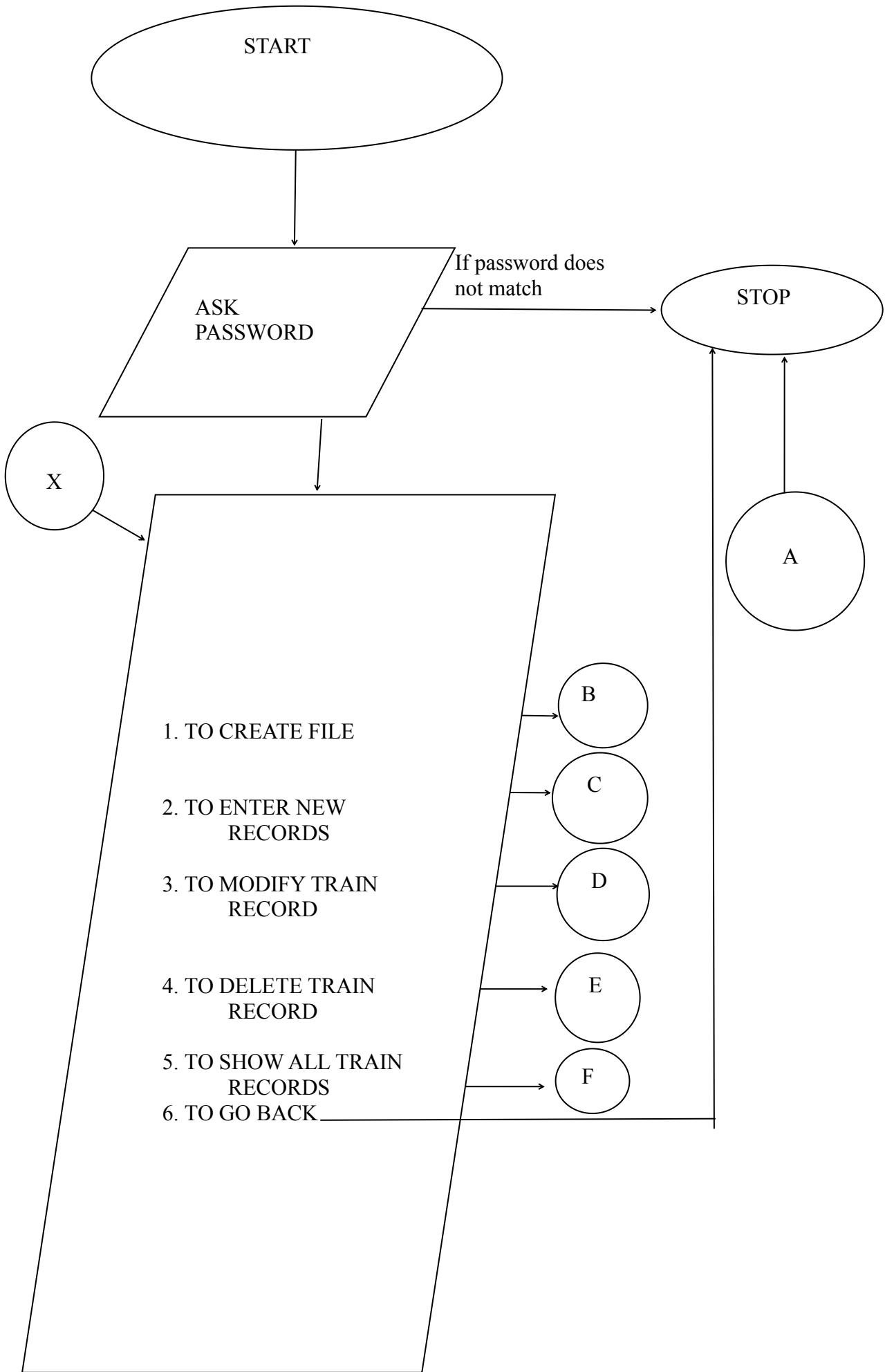


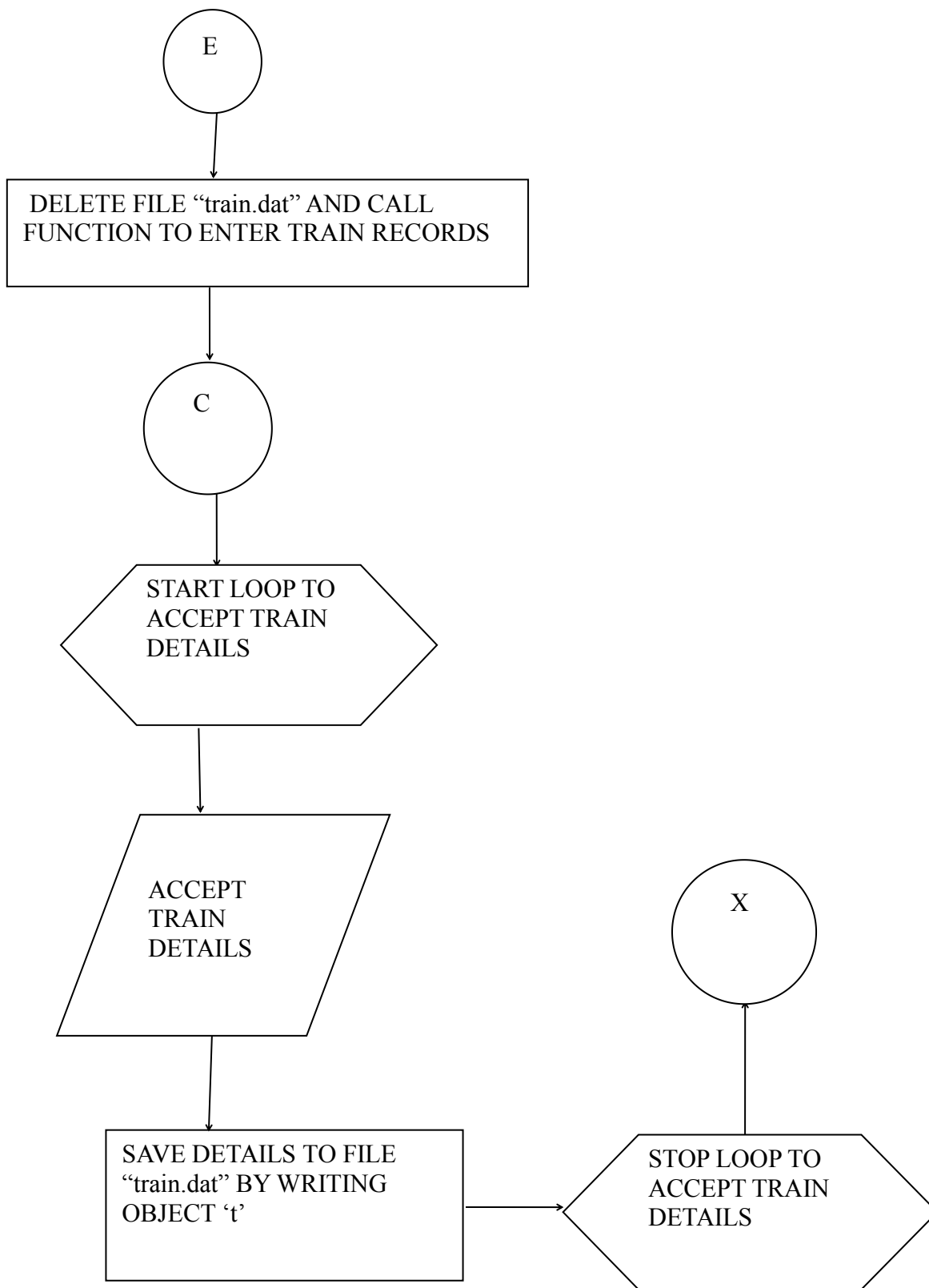
### 3. CANCELLATION

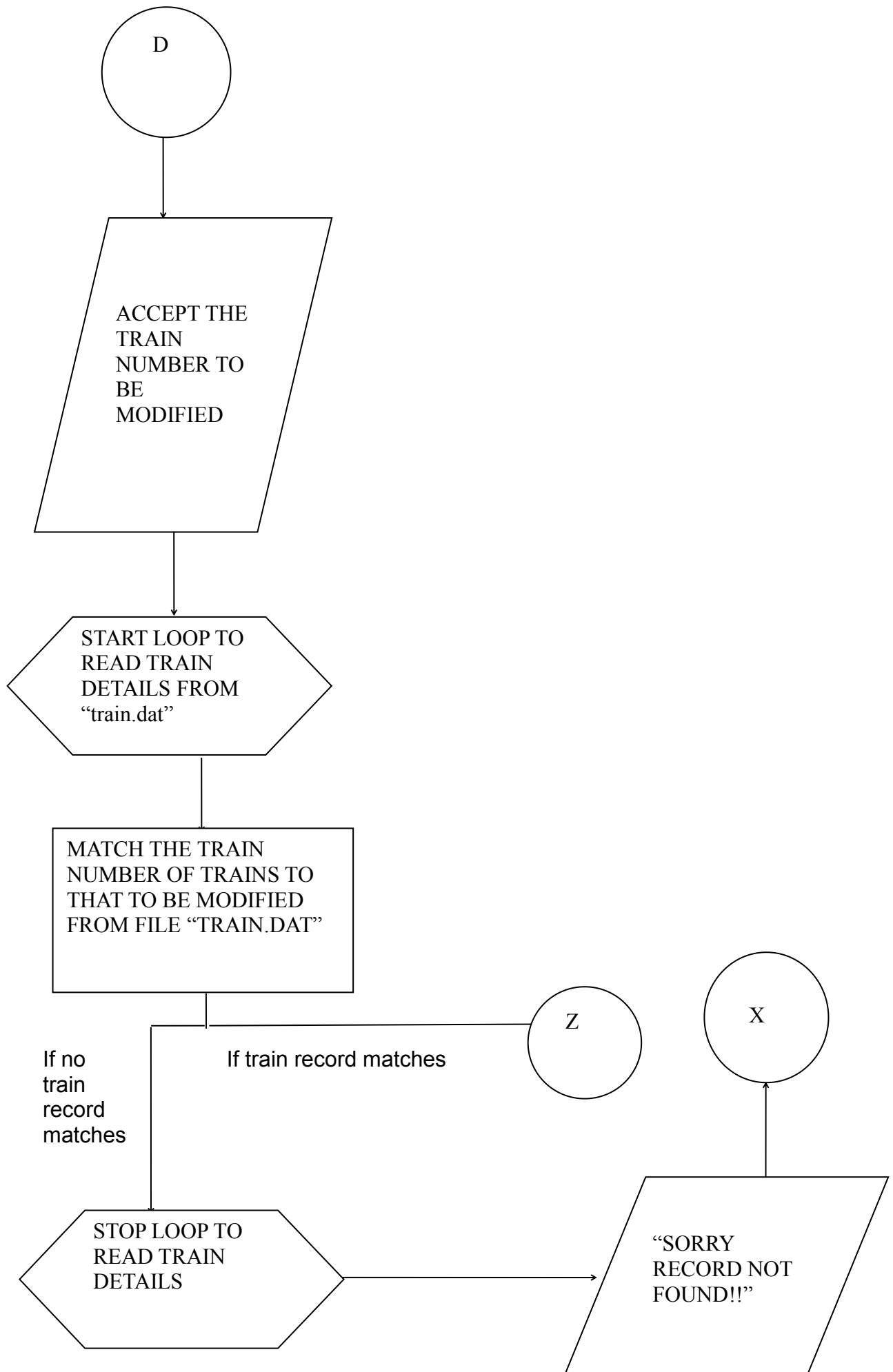


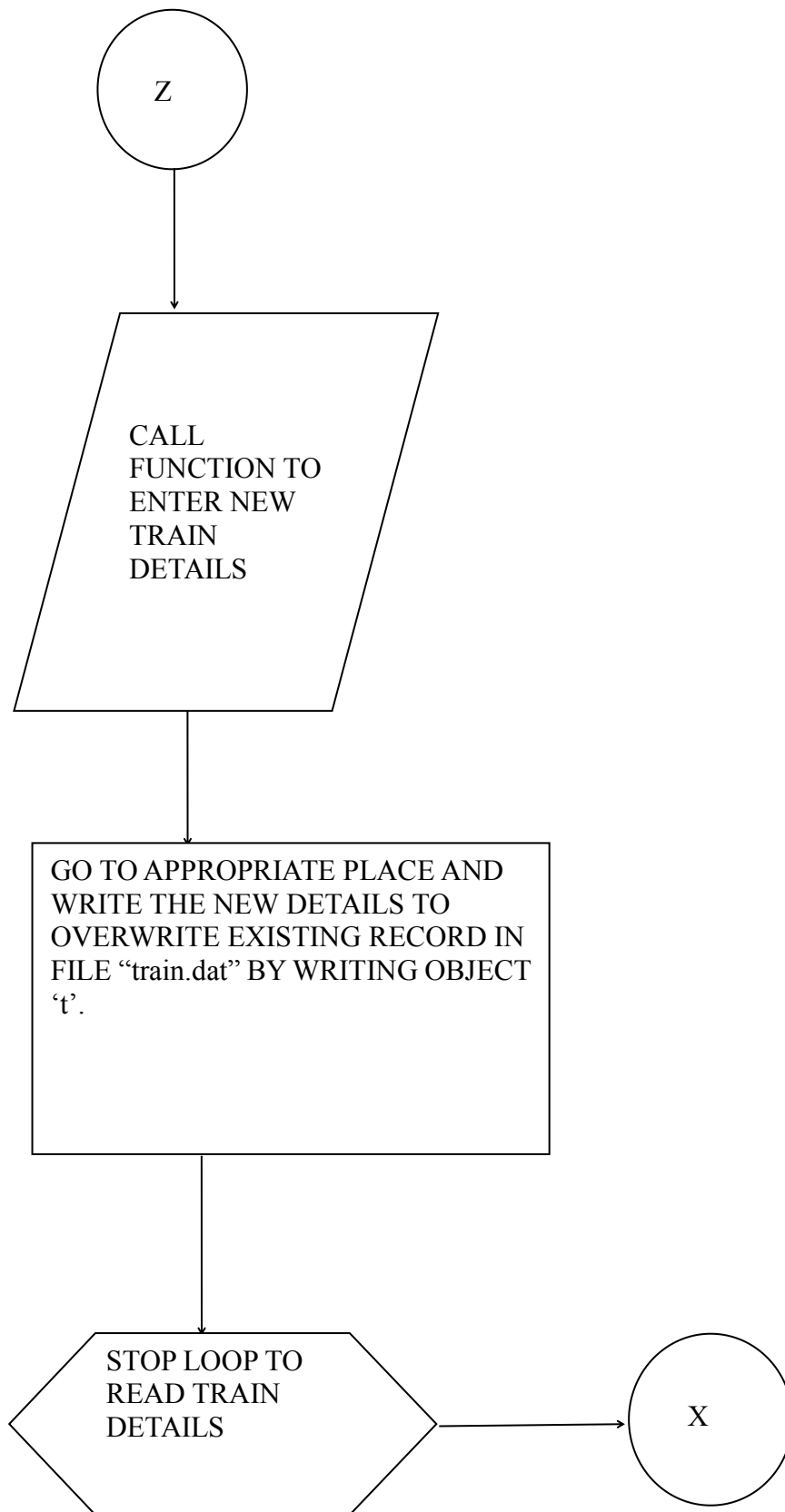


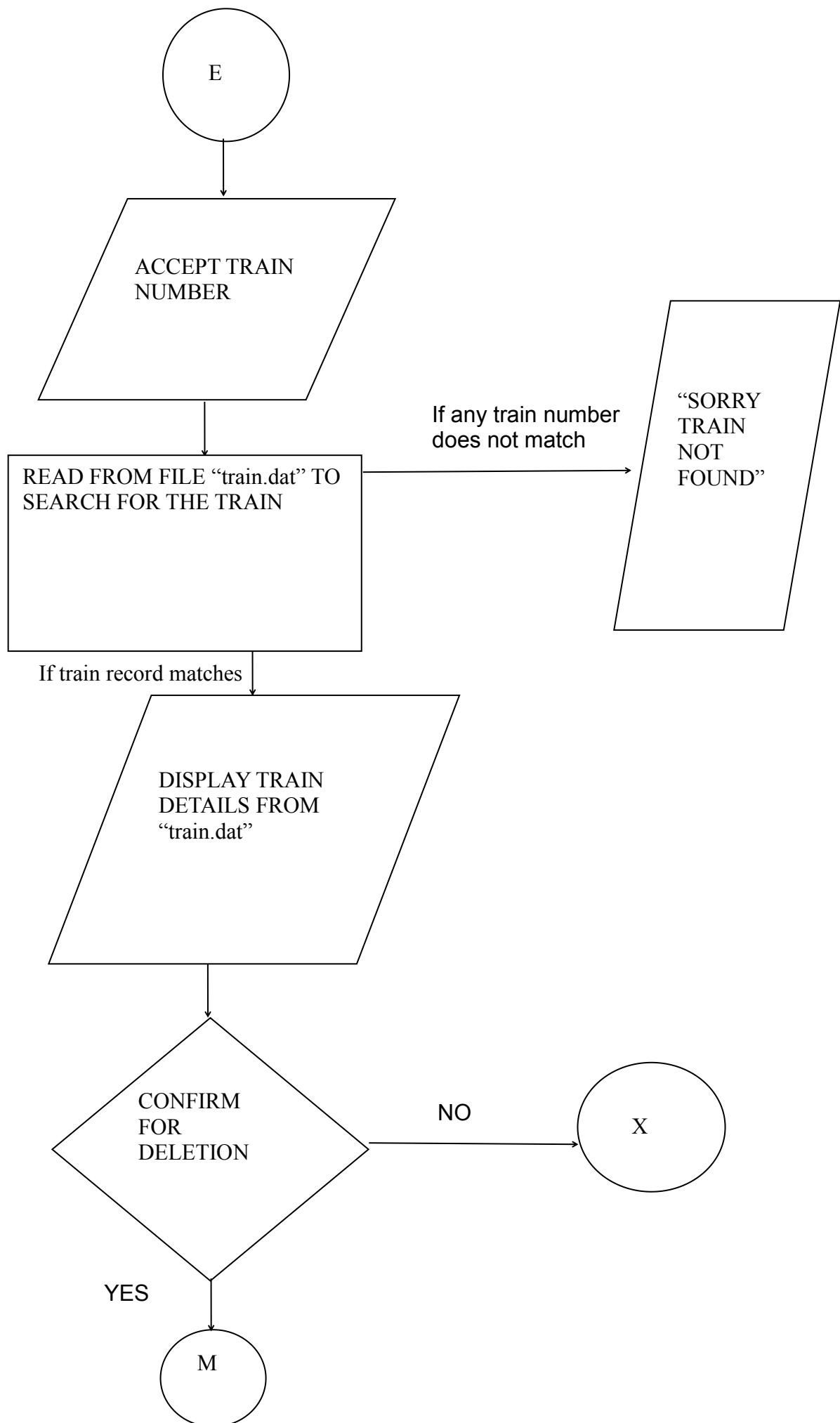
# 4. ADMIN SECTION

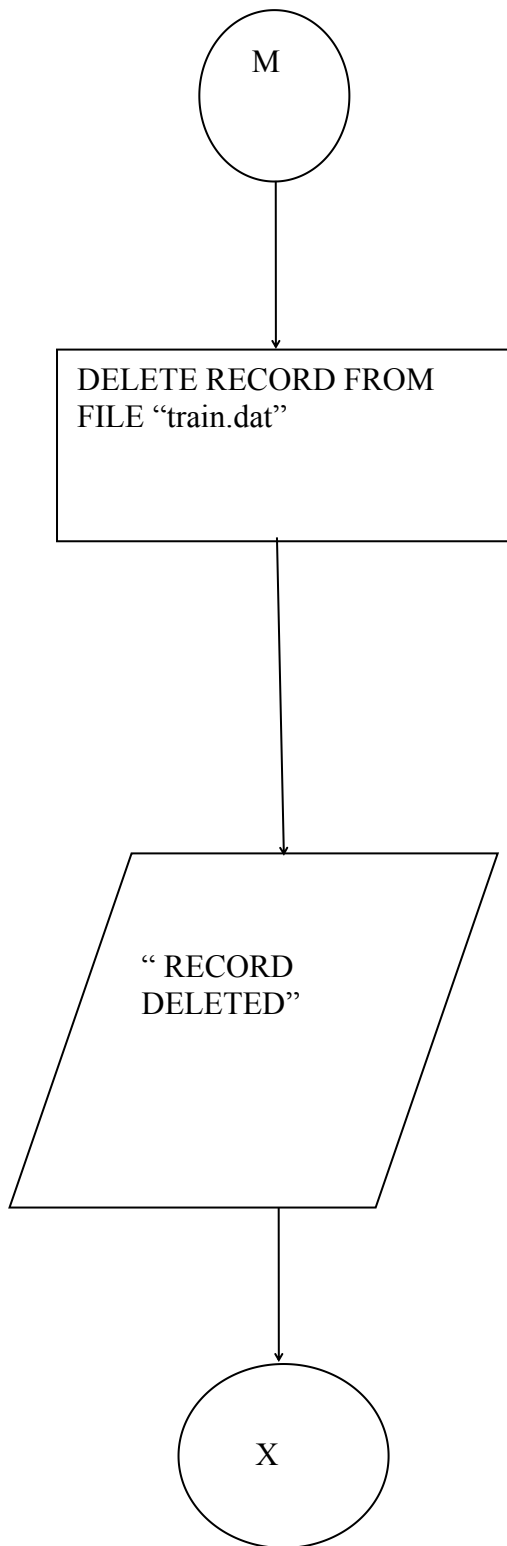






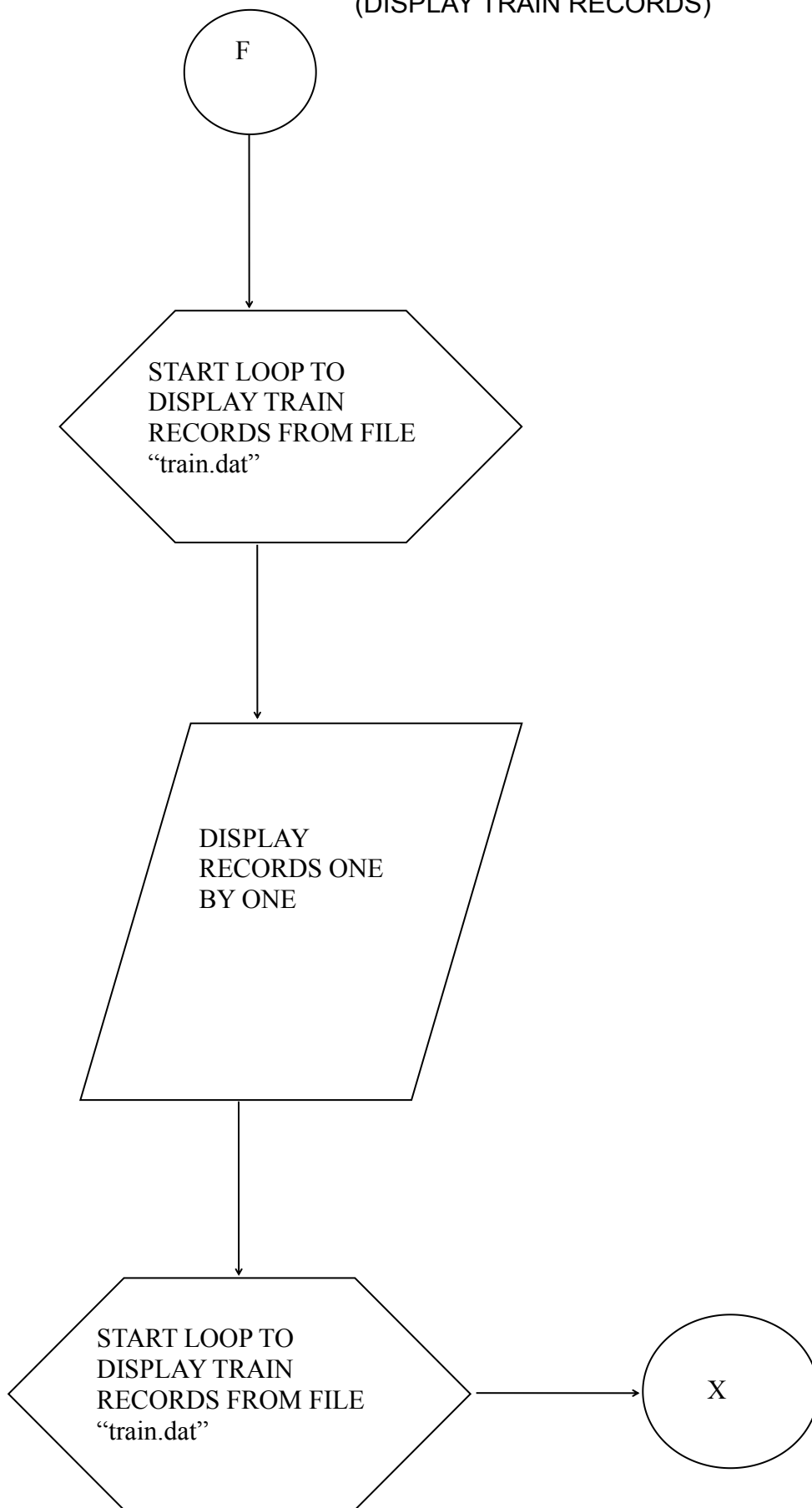








(DISPLAY TRAIN RECORDS)



# CODING

```
#include<fstream.h>
#include<conio.h>
#include<stdio.h>
#include<process.h>
#include<string.h>
```

```
class pn
{
```

```
public:
```

```
int pnr;
```

```
}x;
```

```
void star();    // FUNCTION TO PRINT STARS ON
                // SCREEN
```

```
class train
{
```

```
public:
```

```
int tno,ac,sleeper,sc,time,dist,fac,fsleeper,fsc;
char from[100],to[100],name[100];
```

```
public:
```

```

void modify();      // FUNCTION TO MODIFY TRAIN
                   // RECORDS
void del();         // FUNCTION TO DELETE TRAIN
                   // RECORDS
void display();     // FUNCTION TO DISPLAY TRAIN
                   // RECORDS
void acp();         // FUNCTION TO ACCEPT TRAIN
                   // RECORDS
void check();       // FUNCTION TO CHECK AVAILABILITY

void update(int);  // FUNCTION TO UPDATE NO. OF
                   // SEATS AFTER CANCELLATION

```

```

int ret()  // ACCESSOR FUNCTION TO RETURN TRAIN
          //NUMBER
{
return tno;
}

```

```

char *f1()  // ACCESSOR FUNCTION TO RETURN
            // STARTING POINT
{
return from;
}

```

```

char *t1()  // ACCESSOR FUNCTION TO RETURN
            //DESTINATION
{
return to;
}

```

```

}t;      // END OF CLASS

```

```

void add();  // FUNCTION TO ADD TRAIN RECORDS

```

```

void disp(); // FUNCTION TO DISPLAY TRAIN RECORDS

```

```

class book
{

public:

int no,tickets,pnr;
char pto[100],pfrom[100],clas[20];
long int amt;


int retpnr() // ACCESSOR FUNCTION TO RETURN PNR
            //NUMBER
{
return pnr;

}

int rettic() // ACCESSOR FUNCTION TO RETURN TICKETS
{
return tickets;

}

int retamt() // ACCESSOR FUNCTION TO RETURN AMOUNT
{
return amt;

}

void get();      // FUNCTION TO GET BOOKING DETAILS
int assign();    // FUNCTION TO CALCULATE AND ASSIGN
                // BOOKING DETAILS
void show();     // FUNCTION TO DISPLAY BOOKING
                // DETAILS
void cancel();   // FUNCTION TO CANCEL BOOKING


}b;              // END OF CLASS

```

```

class passenger
{

public:

int pnr,page;
char psex,pname[100];


void getp();          // FUNCTION TO GET PASSENGER
                      // DETAILS
void showp(int);      // FUNCTION TO SHOW PASSENGER
                      // DETAILS


}p; // END OF CLASS


void admin(); // ADMIN SECTION TO MANIPULATE TRAIN
              // RECORDS


void main() // STARTING MAIN FUNCTION
{
int n;

while(1)
{
clrscr();
star();

```

```

gotoxy(30,2);cout<<" WELCOME TO INDIAN RAILWAYS";
gotoxy(5,5); cout<<" ENTER 1 TO CHECK AVAILABILITY ";
gotoxy(5,6); cout<<" ENTER 2 TO BOOK TICKETS";
gotoxy(5,7); cout<<" ENTER 3 FOR CANCELLATION";
gotoxy(5,8); cout<<" ENTER 4 FOR ADMIN ACCESS";
gotoxy(5,9); cout<<" ENTER 5 TO EXIT";
gotoxy(5,14);cout<<" ENTER YOUR CHOICE      -      ";
cin>>n;

```

```

clrscr();

```

```

switch (n)
{

```

```

    case 1 : t.check();
             getch();
             break;

```

```

    case 2 : b.get();
             getch();
             break;

```

```

    case 3 : b.cancel();
             getch();
             break;

```

```

    case 4 : admin();
             break;

```

```

    case 5 : exit(0);

```

```

    default: clrscr();
             star();
             gotoxy(10,10);cout<<" WRONG CHOICE";

```

```

        getch();

    }    // END OF SWITCH

}    // END OF WHILE

}    // END OF MAIN

```

```

void add() // FUNCTION TO WRITE TRAIN RECORDS
{
    clrscr();
    star();
    fstream rr;
    rr.open("train.dat",ios::app|ios::in|ios::binary);

    char ch;

    do
    {

        clrscr();
        star();
        t.acp();
        rr.write((char*)&t,sizeof t);

        gotoxy(10,15);cout<<" ENTER MORE RECORDS ? (Y/N) -
                                ".
                                ";
        cin>>ch;

    }while(ch=='y'||ch=='Y');

    rr.close();

}    // END OF FUNCTION

```

```

void disp() // FUNCTION TO DISPLAY TRAIN RECORDS
{
fstream rr;
rr.open("train.dat",ios::in|ios::binary);

while(rr.read((char*)&t,sizeof t))
{
clrscr();
star();
t.display();
}

rr.close();

} // END OF FUNCTION

```

```

void train::display() // FUNCTION TO DISPLAY TRAIN
//RECORD
{
star();

gotoxy(5,3);cout<<" TRAIN NUMBER           "<<tno;
gotoxy(5,4);cout<<" TRAIN NAME
                "<<name;
gotoxy(5,5);cout<<" STARTING POINT
                "<<from;
gotoxy(5,6);cout<<" DESTINATION           "<<to;
gotoxy(5,7);cout<<" NUMBER OF AC TICKETS AVAILABLE
                "<<ac;
gotoxy(5,8);cout<<" PRICE OF EACH AC TICKET      Rs.
                "<<fac;
gotoxy(5,9);cout<<" NUMBER OF SLEEPER TICKETS
                AVAILABLE      "<<sleeper;
gotoxy(5,10);cout<<" PRICE OF EACH SLEEPER CLASS
                TICKET   Rs.      "<<fsleeper;

```



```

gotoxy(5,11);cout<<" NUMBER OF SECOND CLASS TICKETS
                AVAILABLE "<<sc;
gotoxy(5,12);cout<<" PRICE OF EACH SECOND CLASS
                TICKET  Rs.    "<<fsc;
gotoxy(5,13);cout<<" DURATION OF JOURNEY (hrs)
                "<<time;
gotoxy(5,14);cout<<" TOTAL DISTANCE (km)
                "<<dist;

getch();
}

```

```

void train::acp() // FUNCTION TO ACCEPT TRAIN
                // RECORDS
{

```

```

clrscr();
star();

```

```

gotoxy(5,3);cout<<" ENTER TRAIN NUMBER
                ";cin>>tno;
gotoxy(5,4);cout<<" ENTER TRAIN NAME
                ";gets(name);
gotoxy(5,5);cout<<" ENTER STARTING POINT
                ";gets(from);
gotoxy(5,6);cout<<" ENTER DESTINATION
                ";gets(to);
gotoxy(5,7);cout<<" ENTER NUMBER OF AC TICKETS TO BE
                MADE AVAILABLE      ";cin>>ac;
gotoxy(5,8);cout<<" ENTER PRICE OF EACH AC TICKET
                Rs.    ";cin>>fac;
gotoxy(5,9);cout<<" ENTER NUMBER OF SLEEPER TICKETS
                TO BE MADE AVAILABLE
                ";cin>>sleeper;
gotoxy(5,10);cout<<" ENTER PRICE OF EACH SLEEPER
                TICKET          Rs.
                ";cin>>fsleeper;
gotoxy(5,11);cout<<" ENTER NUMBER OF SECOND CLASS
                TICKETS TO BE MADE AVAILABLE
                ";cin>>sc;

```

```

gotoxy(5,12);cout<<" ENTER PRICE OF EACH SECOND
                        CLASS TICKET           Rs. ";cin>>fsc;
gotoxy(5,13);cout<<" ENTER DURATION OF JOURNEY (hrs)
                        ";cin>>time;
gotoxy(5,14);cout<<" ENTER TOTAL DISTANCE (km)
                        ";cin>>dist;

} // END OF FUNCTION

```

```

void train::modify() // FUNCTION TO MODIFY TRAIN
                    //RECORD

```

```

{
clrscr();
star();

```

```

fstream rr;
rr.open("train.dat",ios::in|ios::out|ios::binary);
int id,f=0,n=0,p;

```

```

gotoxy(5,3);cout<<" ENTER TRAIN NUMBER OF TRAIN TO
                        BE MODIFIED - ";
cin>>id;

```

```

while(rr.read((char*)&t,sizeof t))
{
n++;

```

```

if(id==t.ret())
{
acp();
p=(n-1)*(sizeof t);
rr.seekp(p,ios::beg);
rr.write((char*)&t,sizeof t);
f=1;
}

```

```

}

```

```

if(f==0)
{
gotoxy(10,6);cout<<" SORRY RECORD NOT FOUND !!";
}

```

```

rr.close();

```

```

}    //END OF FUNCTION

```

```

void train::del() // FUNCTION TO DELETE TRAIN RECORDS
{
clrscr();
star();

```

```

int id,f=0;

```

```

fstream rr;
rr.open("train.dat",ios::in|ios::binary);

```

```

fstream t1;
t1.open("temp.dat",ios::out|ios::binary);

```

```

gotoxy(5,2);cout<<" ENTER TRAIN NUMBER OF TRAIN TO
                    BE DELETED    -  ";

```

```

cin>>id;

```

```

while(rr.read((char*)&t,sizeof t))
{    // 1

```

```

if(id==t.ret())
{    // 2
char ch;
f=1;
t.display();

```

```

gotoxy(5,17);cout<<" ARE YOU SURE ? (Y/N)    -    ";
cin>>ch;

```

```

if(ch=='y' || ch=='Y')
{
gotoxy(10,19);cout<<" RECORD DELETED";
}

else
{
t1.write((char*)&t,sizeof t);
}

}      // 2

else
t1.write((char*)&t,sizeof t);

}      // 1

if(f==0)
{
gotoxy(10,10);cout<<" SORRY TRAIN NOT FOUND!!";
}

rr.close();
t1.close();

remove("train.dat");
rename("temp.dat","train.dat" );

} // END OF FUNCTION


void train::check()    // FUNCTION TO CHECK TICKET
                      // AVAILABILITY
{
star();

char T[100],f[100];

```

```
int flag=0;
```

```
gotoxy(10,5);cout<<" ENTER THE STARTING POINT    -    ";  
gets(f);
```

```
gotoxy(10,8);cout<<" ENTER THE DESTINATION        -    ";  
gets(T);
```

```
fstream rr;  
rr.open("train.dat",ios::in|ios::binary);
```

```
while(rr.read((char*)&t,sizeof t))  
{
```

```
    if( strcmpi(f,t.f1())==0 && strcmpi(T,t.t1())==0 )  
    {
```

```
        clrscr();  
        star();  
        flag=1;  
        t.display();  
    }
```

```
}          // END OF WHILE
```

```
if(flag==0)  
{  
gotoxy(10,14);cout<<" SORRY NO AVAILABLE TRAINS  
                        FOUND ";  
}
```

```
}          //END OF FUNCTION
```

```
void train::update(int q) // FUNCTION TO UPDATE TICKETS  
                        // AFTER CANCELLATION  
{
```

```

fstream rr;
int n,tic;

rr.open("booking.dat",ios::in|ios::out|ios::binary);

while(rr.read((char*)&b,sizeof b))
{

if(b.pnr==q)
{

n=b.no;
tic=b.tickets;

rr.close();

rr.open("train.dat",ios::in|ios::out|ios::binary);
rr.seekg(0,ios::beg);

int f=0,p;

while(rr.read((char*)&t,sizeof t))
{

f++;

if(t.tno==n) //1
{
p=(f-1)*(sizeof (t));
rr.seekp(p,ios::beg);

if(strncmpi(b.clas,"ac")==0)
{

t.ac+=tic;
rr.write((char*)&t,sizeof t);
}

else if(strncmpi(b.clas,"sleeper")==0)

```

```

{
t.sleeper+=tic;
rr.write((char*)&t,sizeof t);
}

else
{
t.sc+=tic;
rr.write((char*)&t,sizeof t);
}

}          //1

}          // END OF WHILE 2

rr.close();

break;
}          // END OF IF

}          // END OF WHILE 1

}          // END OF FUNCTION

```

```

void book::get() // FUNCTION TO BOOK TICKETS

```

```

{
int n=1;

while(n==1)
{          // WHILE 1
clrscr();

```

```
star();
```

```
gotoxy(33,3);cout<<" WELCOME TO BOOKING MENU";  
gotoxy(5,6);cout<<" ENTER 1 TO BOOK TICKETS";  
gotoxy(5,8);cout<<" ENTER 2 TO GO BACK";  
gotoxy(5,14);cout<<" ENTER YOUR CHOICE - ";  
cin>>n;
```

```
if(n==1)
```

```
{ // IF 1
```

```
clrscr();  
star();
```

```
int flag=0;
```

```
gotoxy(10,3);cout<<" ENTER THE BOARDING POINT - ";  
gets(pfrom);
```

```
gotoxy(10,6);cout<<" ENTER THE DESTINATION - ";  
gets(pto);
```

```
fstream rr;  
rr.open("train.dat",ios::in|ios::binary);
```

```
fstream pas;  
pas.open("passenger.dat",ios::app|ios::binary);
```

```
rr.seekg(0,ios::beg);
```

```
while(rr.read((char*)&t,sizeof t))  
{ // WHILE 2
```

```
if( strcmpi(pfrom,t.f1())==0 && strcmpi(pto,t.t1())==0 )  
{  
clrscr();  
star();
```



```

flag=1;
t.display();
}

} // END OF WHILE 2


rr.close();

if(flag==0)
{
gotoxy(10,14);cout<<" SORRY NO AVAILABLE TRAINS
                        FOUND ";
getch();
}

// STARTING THE BOOKING PROCESS


if(flag==1)
{    // IF 2

int f1=0;

clrscr();
star();

gotoxy(5,5);cout<<" ENTER THE TRAIN NUMBER OF WHICH
                        YOU WANT TO BOOK TICKETS    -
                        ";cin>>no;

rr.open("train.dat",ios::in|ios::binary);
rr.seekg(0,ios::beg);

while(rr.read((char*)&t,sizeof t) )
{    // WHILE 3

if(no==t.ret())
{    // IF 3

f1=1;

```

```

gotoxy(5,6);cout<<" ENTER THE NUMBER OF TICKETS YOU
                        WANT TO BOOK (MAX. 6)      -   ";
cin>>tickets;

if(tickets>6)
{
gotoxy(5,10);cout<<" SORRY CANNOT BOOK MORE THAN 6
                        TICKETS";

getch();
break;
}

gotoxy(5,7);cout<<" ENTER THE CLASS  ( AC / SLEEPER /
                        SECOND CLASS (SC) )      -   ";
gets(clas);

if(strcmpi(clas,"ac")!=0 && strcmpi(clas,"sleeper")!=0 &&
strcmpi(clas,"sc")!=0)
{
gotoxy(5,10);cout<<" SORRY YOU HAVE ENTERED
                        INCORRECT CLASS";

getch();
break;
}

int z;

z=b.assign(); // CALLING THE FUNCTION TO ASSIGN
              // BOOKING DETAILS

if(z==0)
{
gotoxy(5,9);cout<<" SORRY CANNOT BOOK "<<tickets<<"
                        TICKETS";

getch();
break;
}

```

```

else
{

for(int i=1;i<=tickets;i++)
{
clrscr();
star();

gotoxy(5,2);cout<<" ENTER DETAILS OF PASSENGER  "<<i;
p.getp();
p.pnr=b.pnr;

pas.write((char*)&p,sizeof p);
}

pas.close();

b.show();
p.showp(b.pnr);
getch();

}

} // END OF IF 3


} // END OF WHILE 3

if (f1==0)
{
gotoxy(10,10);cout<<" SORRY YOU HAVE ENTERED
                                INCORRECT TRAIN NUMBER ";

getch();
}

} // END OF IF 2

```

```

pas.close();
rr.close();

} // END OF IF 1

} // END OF WHILE 1


} // END OF FUNCTION


int book::assign() // FUNCTION TO CALCULATE AND
                  // ASSIGN BOOKING DETAILS
{

int n=0,p;

fstream rr;
rr.open("train.dat",ios::in|ios::out|ios::binary);

rr.seekg(0,ios::beg);

while(rr.read((char*)&t,sizeof t))
{
n++;

if(no==t.ret())
{

if(strcmpi(clas,"ac")==0)
{
amt=tickets*t.fac;

p=(n-1)*sizeof (t);
rr.seekp(p,ios::beg);

```

```

t.ac=t.ac-tickets;

if(t.ac<0)
{
return 0;
}

else
rr.write((char*)&t,sizeof t);
}

else if(strcmpi(clas,"sleeper")==0)
{
amt=tickets*t.fsleeper;

p=(n-1)*sizeof (t);
rr.seekp(p,ios::beg);

t.sleeper=t.sleeper-tickets;

if(t.sleeper<0)
return 0;

else
rr.write((char*)&t,sizeof t);
}

else
{
amt=tickets*t.fsc;

p=(n-1)*sizeof (t);
rr.seekp(p,ios::beg);

t.sc=t.sc-tickets;

if(t.sc<0)
return 0;

```

```

else
rr.write((char*)&t,sizeof t);

}

break;
}

}

fstream b1;
b1.open("booking.dat",ios::app|ios::binary);

fstream rr2;
rr2.open("pnr.dat",ios::in|ios::binary);

if(!rr2)
{
rr2.close();

rr2.open("pnr.dat",ios::in|ios::app|ios::binary);
rr2.seekp(0,ios::beg);

x.pnr=1000;

rr2.write((char*)&x,sizeof x);
rr2.close();
}

else
{
int d;

rr2.seekg(0,ios::beg);

while(rr2.read((char*)&x,sizeof x))
{

d=x.pnr;

```

```
}
```

```
rr2.close();  
fstream r2;  
r2.open("pnr.dat",ios::app|ios::binary);
```

```
x.pnr=++d;
```

```
r2.write((char*)&x,sizeof x);  
r2.close();  
}
```

```
b.pnr=x.pnr;
```

```
b1.write((char*)&b,sizeof b);  
b1.close();
```

```
return 1;
```

```
} //END OF FUNCTION
```

```
void book::show()    // FUNCTION TO DISPLAY BOOKING  
                    //DETAILS
```

```
{
```

```
clrscr();
```

```
star();
```

```
gotoxy(33,2);cout<<" TICKET";  
gotoxy(5,4); cout<<" PNR NUMBER          -    "<<pnr;  
gotoxy(5,5); cout<<" TRAIN NUMBER        -    "<<no;  
gotoxy(5,6); cout<<" TOTAL PASSENGERS    -    "<<tickets;  
gotoxy(5,7); cout<<" TOTAL AMOUNT        -    "<<amt;  
gotoxy(5,8); cout<<" BOARDING POINT    -    "<<pfrom;  
gotoxy(5,9); cout<<" DESTINATION      -    "<<pto;
```

```
gotoxy(8,23);cout<<" YOUR TICKET HAS BEEN RESERVED";  
} // END OF FUNCTION
```

```
void book::cancel() // FUNCTION TO CANCEL BOOKING  
{  
    star();
```

```
    train tr;
```

```
    fstream b1;  
    b1.open("booking.dat",ios::in|ios::binary);
```

```
    fstream rr;  
    rr.open("passenger.dat",ios::in|ios::binary);
```

```
    fstream t;  
    t.open("temp",ios::out|ios::binary);
```

```
    fstream tmp;  
    tmp.open("t2",ios::out|ios::binary);
```

```
    int id;  
    int flag=0,f=0;  
    char ch;
```

```
    gotoxy(5,5);cout<<" ENTER YOUR PNR NUMBER    -    ";  
    cin>>id;
```

```
    while(b1.read((char*)&b,sizeof b))  
    {
```

```
        if(id==b.retpnr())  
        {  
            clrscr();  
            star();
```

```
            flag=1;
```



```

f=1;
gotoxy(5,2);cout<<" PNR NUMBER          -
    "<<b.retpnr();
gotoxy(5,3);cout<<" TOTAL TICKETS      - "<<b.rettic();
gotoxy(5,4);cout<<" TOTAL AMOUNT      - "<<b.retamt();
gotoxy(5,5); cout<<" TOTAL AMOUNT      - "<<amt;
gotoxy(5,6); cout<<" BOARDING POINT    - "<<pfrom;
gotoxy(5,7); cout<<" DESTINATION      - "<<pto;
p.showp(id);

```

```

gotoxy(5,22);cout<<" ARE YOU SURE ? (Y/N) - ";
cin>>ch;

```

```

if(ch=='y' || ch=='Y')
{
gotoxy(5,23);cout<<" YOUR BOOKING HAS BEEN
                        CANCELLED";
tr.update(id);
}

```

```

else
{
flag=0;
t.write((char*)&b,sizeof b);
}

```

```

}

```

```

else
{
t.write((char*)&b,sizeof b);
}

```

```

}

```

```

remove("booking.dat");

```

```
rename("temp","booking.dat");
```

```
if(flag==1)
```

```
{
```

```
    rr.seekg(0,ios::beg);
```

```
    while(rr.read((char*)&p,sizeof p))  
    {
```

```
        if(id!=b.retpnr())  
            tmp.write((char*)&p,sizeof p);
```

```
    }
```

```
remove("passenger.dat");
```

```
rename("t2","passenger.dat");
```

```
}
```

```
if(f==0)
```

```
{
```

```
gotoxy(5,8);cout<<" SORRY INVALID PNR  ";
```

```
}
```

```
rr.close();
```

```
tmp.close();
```

```
b1.close();
```

```
t.close();
```

```
} // END OF FUNCTION
```

```
void passenger::getp()  
{  
    star();
```

```
    gotoxy(5,3);cout<<" ENTER NAME    - ";gets(pname);  
    gotoxy(5,4);cout<<" ENTER AGE      - ";cin>>page;  
    gotoxy(5,5);cout<<" ENTER SEX (M/F) -    ";cin>>psex;
```

```
} // END OF FUNCTION
```

```
void passenger::showp(int q) // FUNCTION TO SHOW  
                             // PASSENGER DETAILS
```

```
{
```

```
    int i=14;
```

```
    fstream rr;  
    rr.open("passenger.dat",ios::in|ios::binary);
```

```
    gotoxy(30,12);cout<<" PASSENGER LIST";
```

```
    gotoxy(5,13);cout<<" PASSENGER NAME  -  PASSENGER  
    AGE    -  PASSENGER SEX  ";
```

```
    while(rr.read((char*)&p,sizeof p))  
    {
```

```

        if(p.pnr==q)
        {
            gotoxy(6,++i);cout<<pname;
            gotoxy(30,i);cout<<page;
            gotoxy(50,i);cout<<psex;
        }

    }

    rr.close();


} // END OF FUNCTION


void admin() // FUNCTION TO MANIPULATE TRAIN
            // RECORDS
{
    star();

    int m;
    char p[20];

    gotoxy(10,5);cout<<" ENTER THE PASSWORD    -    ";

    for(int i=0;i<20;i++)
    {
        p[i]=getch();

        if(p[i]==13)
        {
            p[i]='\0';
            break;
        }

        else
            cout<<"*";
    }
}

```

```

if(strcmp(p,"railway123")==0)
{

do
{
clrscr();
star();

gotoxy(33,3);cout<<" ADMIN SECTION";
gotoxy(5,5); cout<<" ENTER 1 TO CREATE NEW FILE";
gotoxy(5,6); cout<<" ENTER 2 TO ENTER NEW RECORD ";
gotoxy(5,7); cout<<" ENTER 3 TO MODIFY A TRAIN
                        RECORD";
gotoxy(5,8); cout<<" ENTER 4 TO DELETE A TRAIN
                        RECORD";
gotoxy(5,9); cout<<" ENTER 5 TO DISPLAY ALL THE
                        CURRENT RECORDS";
gotoxy(5,10); cout<<" ENTER 6 TO GO BACK";
gotoxy(5,15); cout<<" ENTER YOUR CHOICE      -      ";
cin>>m;

switch (m)
{
    case 1: remove("train.dat");
            add();
            getch();
            break;

    case 2: add();
            getch();
            break;

    case 3: t.modify();
            getch();
            break;

    case 4: t.del();

```

```

        getch();
        break;

    case 5: disp();
        getch();
        break;

    }

}while(m>=1&&m<=5);

} // END OF IF

else
{
    clrscr();
    star();
    gotoxy(10,7);cout<<" WRONG PASSWORD";
    gotoxy(10,10);cout<<" ACCESS DENIED";
    getch();
}

} // END OF FUNCTION


void star() // FUNCTION TO PRINT STARS ON SCREEN
{

    for(int i=1;i<81;i++)
    {
        gotoxy(i,1);cout<<"*";
        gotoxy(i,24);cout<<"*";
    }

    for(i=1;i<25;i++)
    {

```

```
gotoxy(1,i);cout<<"*";  
gotoxy(80,i);cout<<"*";  
}  
  
} // END OF FUNCTION
```

# OUTPUTS

[illegible][illegible]



```

*****
*
*
*
*   ENTER THE PASSWORD   -   *****
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*
*
*   ADMIN SECTION
*
*   ENTER 1 TO CREATE NEW FILE
*   ENTER 2 TO ENTER NEW RECORD
*   ENTER 3 TO MODIFY A TRAIN RECORD
*   ENTER 4 TO DELETE A TRAIN RECORD
*   ENTER 5 TO DISPLAY ALL THE CURRENT RECORDS
*   ENTER 6 TO GO BACK
*
*
*
*
*
*
*
*
*
*
*   ENTER YOUR CHOICE   -   1_
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                      1000
*   ENTER TRAIN NAME                        JAIPUR EXPRESS
*   ENTER STARTING POINT                   DELHI
*   ENTER DESTINATION                     JAIPUR
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 60
*   ENTER PRICE OF EACH AC TICKET          Rs. 800
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH SLEEPER TICKET     Rs. 590
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 60
*   ENTER PRICE OF EACH SECOND CLASS TICKET Rs. 400
*   ENTER DURATION OF JOURNEY (hrs)        5
*   ENTER TOTAL DISTANCE (km)              300
*   ENTER MORE RECORDS ? (Y/N) - Y_
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                      1001
*   ENTER TRAIN NAME                        BOMBAY RAIL
*   ENTER STARTING POINT                   MUMBAI
*   ENTER DESTINATION                     DELHI
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH AC TICKET          Rs. 1100
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH SLEEPER TICKET     Rs. 790
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 40
*   ENTER PRICE OF EACH SECOND CLASS TICKET Rs. 600
*   ENTER DURATION OF JOURNEY (hrs)        8
*   ENTER TOTAL DISTANCE (km)              1358
*   ENTER MORE RECORDS ? (Y/N) - Y
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                      1002
*   ENTER TRAIN NAME                        JAN SHATABDI
*   ENTER STARTING POINT                   HARIDWAR
*   ENTER DESTINATION                     AMRITSAR
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 80
*   ENTER PRICE OF EACH AC TICKET          Rs. 1000
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 90
*   ENTER PRICE OF EACH SLEEPER TICKET     Rs. 850
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 80
*   ENTER PRICE OF EACH SECOND CLASS TICKET Rs. 700
*   ENTER DURATION OF JOURNEY (hrs)        6
*   ENTER TOTAL DISTANCE (km)              411
*   ENTER MORE RECORDS ? (Y/N) - Y
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                      1003
*   ENTER TRAIN NAME                        KASHMIR EXPRESS
*   ENTER STARTING POINT                   KOLKATA
*   ENTER DESTINATION                     JAMMU
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 70
*   ENTER PRICE OF EACH AC TICKET          Rs. 2200
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH SLEEPER TICKET     Rs. 1700
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 60
*   ENTER PRICE OF EACH SECOND CLASS TICKET Rs. 1300
*   ENTER DURATION OF JOURNEY (hrs)        26
*   ENTER TOTAL DISTANCE (km)              2588
*   ENTER MORE RECORDS ? (Y/N) - N
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*                               ADMIN SECTION
*
*   ENTER 1 TO CREATE NEW FILE
*   ENTER 2 TO ENTER NEW RECORD
*   ENTER 3 TO MODIFY A TRAIN RECORD
*   ENTER 4 TO DELETE A TRAIN RECORD
*   ENTER 5 TO DISPLAY ALL THE CURRENT RECORDS
*   ENTER 6 TO GO BACK
*
*
*
*   ENTER YOUR CHOICE           -           2_
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                      1004
*   ENTER TRAIN NAME                        HYDERABAD EXPRESS
*   ENTER STARTING POINT                    DELHI
*   ENTER DESTINATION                      HYDERABAD
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE 60
*   ENTER PRICE OF EACH AC TICKET                      Rs. 900
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH SLEEPER TICKET                      Rs. 700
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE 50
*   ENTER PRICE OF EACH SECOND CLASS TICKET                      Rs. 600
*   ENTER DURATION OF JOURNEY (hrs)                      12
*   ENTER TOTAL DISTANCE (km)                      688
*   ENTER MORE RECORDS ? (Y/N)      -      N
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER OF TRAIN TO BE MODIFIED      -    1004
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER TRAIN NUMBER                                1009
*   ENTER TRAIN NAME                                SOUTHERN EXPRESS
*   ENTER STARTING POINT                            BANGALORE
*   ENTER DESTINATION                               CHENNAI
*   ENTER NUMBER OF AC TICKETS TO BE MADE AVAILABLE  70
*   ENTER PRICE OF EACH AC TICKET                   Rs. 800
*   ENTER NUMBER OF SLEEPER TICKETS TO BE MADE AVAILABLE  80
*   ENTER PRICE OF EACH SLEEPER TICKET               Rs. 650
*   ENTER NUMBER OF SECOND CLASS TICKETS TO BE MADE AVAILABLE  60
*   ENTER PRICE OF EACH SECOND CLASS TICKET          Rs. 500
*   ENTER DURATION OF JOURNEY (hrs)                  5
*   ENTER TOTAL DISTANCE (km)                        358
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*                               ADMIN SECTION
*
*  ENTER 1 TO CREATE NEW FILE
*  ENTER 2 TO ENTER NEW RECORD
*  ENTER 3 TO MODIFY A TRAIN RECORD
*  ENTER 4 TO DELETE A TRAIN RECORD
*  ENTER 5 TO DISPLAY ALL THE CURRENT RECORDS
*  ENTER 6 TO GO BACK
*
*
*
*
*  ENTER YOUR CHOICE          -          4_
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*  ENTER TRAIN NUMBER OF TRAIN TO BE DELETED      -      1001
*  TRAIN NUMBER                                     1001
*  TRAIN NAME                                       BOMBAY RAIL
*  STARTING POINT                                  MUMBAI
*  DESTINATION                                     DELHI
*  NUMBER OF AC TICKETS AVAILABLE                  50
*  PRICE OF EACH AC TICKET                        Rs. 1100
*  NUMBER OF SLEEPER TICKETS AVAILABLE             50
*  PRICE OF EACH SLEEPER CLASS TICKET              Rs. 790
*  NUMBER OF SECOND CLASS TICKETS AVAILABLE        40
*  PRICE OF EACH SECOND CLASS TICKET              Rs. 600
*  DURATION OF JOURNEY (hrs)                       8
*  TOTAL DISTANCE (km)                             1358
*
*
*  ARE YOU SURE ? (Y/N)      -          Y
*
*  RECORD DELETED_
*
*
*
*
*****

```

```

*****
*
*                               ADMIN SECTION
*
*  ENTER 1 TO CREATE NEW FILE
*  ENTER 2 TO ENTER NEW RECORD
*  ENTER 3 TO MODIFY A TRAIN RECORD
*  ENTER 4 TO DELETE A TRAIN RECORD
*  ENTER 5 TO DISPLAY ALL THE CURRENT RECORDS
*  ENTER 6 TO GO BACK
*
*
*
*
*  ENTER YOUR CHOICE           -           5
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*  TRAIN NUMBER                1000
*  TRAIN NAME                  JAIPUR EXPRESS
*  STARTING POINT              DELHI
*  DESTINATION                 JAIPUR
*  NUMBER OF AC TICKETS AVAILABLE 60
*  PRICE OF EACH AC TICKET      Rs. 800
*  NUMBER OF SLEEPER TICKETS AVAILABLE 50
*  PRICE OF EACH SLEEPER CLASS TICKET Rs. 590
*  NUMBER OF SECOND CLASS TICKETS AVAILABLE 60
*  PRICE OF EACH SECOND CLASS TICKET Rs. 400
*  DURATION OF JOURNEY (hrs)    5
*  TOTAL DISTANCE (km)         300_
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   TRAIN NUMBER                1002
*   TRAIN NAME                  JAN SHATABDI
*   STARTING POINT              HARIDWAR
*   DESTINATION                 AMRITSAR
*   NUMBER OF AC TICKETS AVAILABLE 80
*   PRICE OF EACH AC TICKET      Rs. 1000
*   NUMBER OF SLEEPER TICKETS AVAILABLE 90
*   PRICE OF EACH SLEEPER CLASS TICKET Rs. 850
*   NUMBER OF SECOND CLASS TICKETS AVAILABLE 80
*   PRICE OF EACH SECOND CLASS TICKET Rs. 700
*   DURATION OF JOURNEY (hrs)    6
*   TOTAL DISTANCE (km)         411
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   TRAIN NUMBER                1003
*   TRAIN NAME                  KASHMIR EXPRESS
*   STARTING POINT              KOLKATA
*   DESTINATION                 JAMMU
*   NUMBER OF AC TICKETS AVAILABLE 70
*   PRICE OF EACH AC TICKET      Rs. 2200
*   NUMBER OF SLEEPER TICKETS AVAILABLE 50
*   PRICE OF EACH SLEEPER CLASS TICKET Rs. 1700
*   NUMBER OF SECOND CLASS TICKETS AVAILABLE 60
*   PRICE OF EACH SECOND CLASS TICKET Rs. 1300
*   DURATION OF JOURNEY (hrs)    26
*   TOTAL DISTANCE (km)         2588_
*
*
*
*
*
*
*
*
*****

```



```

*****
*
*   TRAIN NUMBER                1009
*   TRAIN NAME                  SOUTHERN EXPRESS
*   STARTING POINT              BANGALORE
*   DESTINATION                 CHENNAI
*   NUMBER OF AC TICKETS AVAILABLE 70
*   PRICE OF EACH AC TICKET      Rs. 800
*   NUMBER OF SLEEPER TICKETS AVAILABLE 80
*   PRICE OF EACH SLEEPER CLASS TICKET Rs. 650
*   NUMBER OF SECOND CLASS TICKETS AVAILABLE 60
*   PRICE OF EACH SECOND CLASS TICKET Rs. 500
*   DURATION OF JOURNEY (hrs)    5
*   TOTAL DISTANCE (km)         358_
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*                               ADMIN SECTION
*
*   ENTER 1 TO CREATE NEW FILE
*   ENTER 2 TO ENTER NEW RECORD
*   ENTER 3 TO MODIFY A TRAIN RECORD
*   ENTER 4 TO DELETE A TRAIN RECORD
*   ENTER 5 TO DISPLAY ALL THE CURRENT RECORDS
*   ENTER 6 TO GO BACK
*
*
*
*
*   ENTER YOUR CHOICE          -      6_
*
*
*
*
*
*
*
*
*****

```



```

*****
*
*   TRAIN NUMBER                1009
*   TRAIN NAME                  SOUTHERN EXPRESS
*   STARTING POINT              BANGALORE
*   DESTINATION                 CHENNAI
*   NUMBER OF AC TICKETS AVAILABLE 70
*   PRICE OF EACH AC TICKET      Rs. 800
*   NUMBER OF SLEEPER TICKETS AVAILABLE 80
*   PRICE OF EACH SLEEPER CLASS TICKET Rs. 650
*   NUMBER OF SECOND CLASS TICKETS AVAILABLE 60
*   PRICE OF EACH SECOND CLASS TICKET Rs. 500
*   DURATION OF JOURNEY (hrs)    5
*   TOTAL DISTANCE (km)         358_
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*                                     WELCOME TO INDIAN RAILWAYS
*
*
*   ENTER 1 TO CHECK AVAILABILITY
*   ENTER 2 TO BOOK TICKETS
*   ENTER 3 FOR CANCELLATION
*   ENTER 4 FOR ADMIN ACCESS
*   ENTER 5 TO EXIT
*
*
*
*
*   ENTER YOUR CHOICE      -      2
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*                               WELCOME TO BOOKING MENU                               *
*
*
*   ENTER 1 TO BOOK TICKETS
*
*   ENTER 2 TO GO BACK
*
*
*
*
*   ENTER YOUR CHOICE           -           1
*
*
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   ENTER THE BOARDING POINT           -           haridwar
*
*
*   ENTER THE DESTINATION              -           AMRITSAR_
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*   TRAIN NUMBER                1002
*   TRAIN NAME                  JAN SHATABDI
*   STARTING POINT              HARIDWAR
*   DESTINATION                 AMRITSAR
*   NUMBER OF AC TICKETS AVAILABLE 80
*   PRICE OF EACH AC TICKET      Rs. 1000
*   NUMBER OF SLEEPER TICKETS AVAILABLE 90
*   PRICE OF EACH SLEEPER CLASS TICKET Rs. 850
*   NUMBER OF SECOND CLASS TICKETS AVAILABLE 80
*   PRICE OF EACH SECOND CLASS TICKET Rs. 700
*   DURATION OF JOURNEY (hrs)    6
*   TOTAL DISTANCE (km)         411_
*
*
*
*
*
*
*
*
*
*
*****

```

```

*****
*
*
*
*   ENTER THE TRAIN NUMBER OF WHICH YOU WANT TO BOOK TICKETS      -   1002
*   ENTER THE NUMBER OF TICKETS YOU WANT TO BOOK (MAX. 6)         -   3
*   ENTER THE CLASS ( AC / SLEEPER / SECOND CLASS (SC) )          -   ac
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*
*****

```

[illegible][illegible]

[illegible]

```

*****
*                                     *
*                               TICKET *
*                                     *
* PNR NUMBER      -      1000      *
* TRAIN NUMBER    -      1002      *
* TOTAL PASSENGERS -      3        *
* BOARDING POINT  -      haridwar  *
* DESTINATION     -      amritsar   *
*                                     *
*                                     *
*                                     *
*                               PASSENGER LIST *
*                                     *
* PASSENGER NAME  -      PASSENGER AGE  -      PASSENGER SEX *
*                                     *
* Anant           17           m          *
* Aastha          17           f          *
* Ajay            14           m_         *
*                                     *
*                                     *
*                                     *
*                                     *
*                                     *
* YOUR TICKET HAS BEEN RESERVED *
*                                     *
*****

```

```
*****
*
*                                     WELCOME TO BOOKING MENU
*
*
* ENTER 1 TO BOOK TICKETS
*
* ENTER 2 TO GO BACK
*
*
*
*
* ENTER YOUR CHOICE           -           2
*
*
*
*
*
*
*
*
*
*
*
*
*
*****
```

```
*****
*                                     WELCOME TO INDIAN RAILWAYS
*
*
* ENTER 1 TO CHECK AVAILABILITY
* ENTER 2 TO BOOK TICKETS
* ENTER 3 FOR CANCELLATION
* ENTER 4 FOR ADMIN ACCESS
* ENTER 5 TO EXIT
*
*
*
*
* ENTER YOUR CHOICE           -           3_
*
*
*
*
*
*
*
*
*
*
*
*
*
*****
```



```

*****
*   PNR NUMBER       -      1000   *
*   TOTAL TICKETS    -      3      *
*   TOTAL AMOUNT     -     3000    *
*   TOTAL PASSENGERS -      3      *
*   TOTAL AMOUNT     -     3000    *
*   BOARDING POINT   -    haridwar *
*   DESTINATION      -    amritsar  *
*                                   *
*                                   *
*                                   *
*           PASSENGER LIST          *
*   PASSENGER NAME   -   PASSENGER AGE   -   PASSENGER SEX   *
*                                   *
*   Anant            17                   m                   *
*   Aastha           17                   f                   *
*   Ajay             14                   m                   *
*                                   *
*                                   *
*                                   *
*   ARE YOU SURE ? (Y/N) -      y      *
*   YOUR BOOKING HAS BEEN CANCELLED      *
*****

```

```

*****
*                                   *
*           WELCOME TO INDIAN RAILWAYS   *
*                                   *
*   ENTER 1 TO CHECK AVAILABILITY        *
*   ENTER 2 TO BOOK TICKETS              *
*   ENTER 3 FOR CANCELLATION             *
*   ENTER 4 FOR ADMIN ACCESS             *
*   ENTER 5 TO EXIT                      *
*                                   *
*                                   *
*                                   *
*   ENTER YOUR CHOICE -      5          *
*                                   *
*                                   *
*                                   *
*                                   *
*                                   *
*                                   *
*                                   *
*                                   *
*****

```

# **BIBLIOGRAPHY**

## **BOOKS -**

- i.) Computer Science (for class XII)  
by – Sumita Arora