**IMPLEMENTING THE BOOKING OF TICKET LOGIC**

The next task in the project is to **book a ticket**. However this task is itself divided into following subtasks:

1. Accept the passenger details with all the validations applied

2.While accepting the details , allow the user to cancel the process in between

3.After all the input has been received , check whether the seats are available or not .

4. If seat is available , then book the ticket and generate the ticket number otherwise print the message **SEATS FULL** .**We assume that we have 30 seats per train per class.**

We have divided the above requirements into multiple functions:

1. **Passenger \* get\_passenger\_details()** : This function will accept all data about the **passenger** and ticket to be booked with all the validations applied. If the input is correct it will return the address of **Passenger structure variable** and if the user cancels the process in between it returns **NULL**.

2. **int check\_train\_no(char \*)**: This function will accept a train number as argument and check whether a train with that number exists or not. If the train exists it **return 1** otherwise it **returns 0**

3. **int get\_booked\_ticket\_count(char \*,char)**:This function will accept a train number and the travelling class (F or S) as argument and count and **return number of tickets** booked in the given train in the given class.

4. **int last\_ticket\_no()**: This function will count and return the **ticket no** of last booked ticket and if no tickets have been booked it will **return 0**

5. **int book\_ticket(Passenger)**: This function will accept the Passenger variable as argument and book the ticket. If booking is successful it will **return 1** otherwise it will **return 0**.

So before proceeding further make the following declarations in the header file **"rlyres.h"**

**Passenger \* get\_passenger\_details();**

**int check\_train\_no(char \*);**

**int get\_booked\_ticket\_count(char \*,char);**

**int last\_ticket\_no();**

**int book\_ticket(Passenger);**

**DESIGNING THE "check\_train\_no()" FUNCTION IN THE FILE "rlyres.c"**

This function will accept a train number as argument and check whether a train with that number exists or not. If the train exists it **return 1** otherwise it **returns 0**

Following is it's prototype:

**int check\_train\_no(char \*);**

This function will do the following:

1. Open the file **alltrains.dat**

2. **Run a loop** and **read each record** of a Train from the file

3. **Compare** the **train number passed as argument** with **the train number of the record read**.

4.If the match occurs then **return 1** other wise **read the next record**.

5.Finally , if all records are read and the train number is not found then **return 0**.

Following is it's code:

***int check\_train\_no(char \*trainno)***

***{***

***FILE \*fp=fopen("d:\\myproject\\alltrains.dat","rb");***

***Train tr;***

***while(fread(&tr,sizeof(tr),1,fp)==1)***

***{***

***if(strcmp(tr.train\_no,trainno)==0)***

***{***

***fclose(fp);***

***return 1;***

***}***

***}***

***fclose(fp);***

***return 0;***

***}***

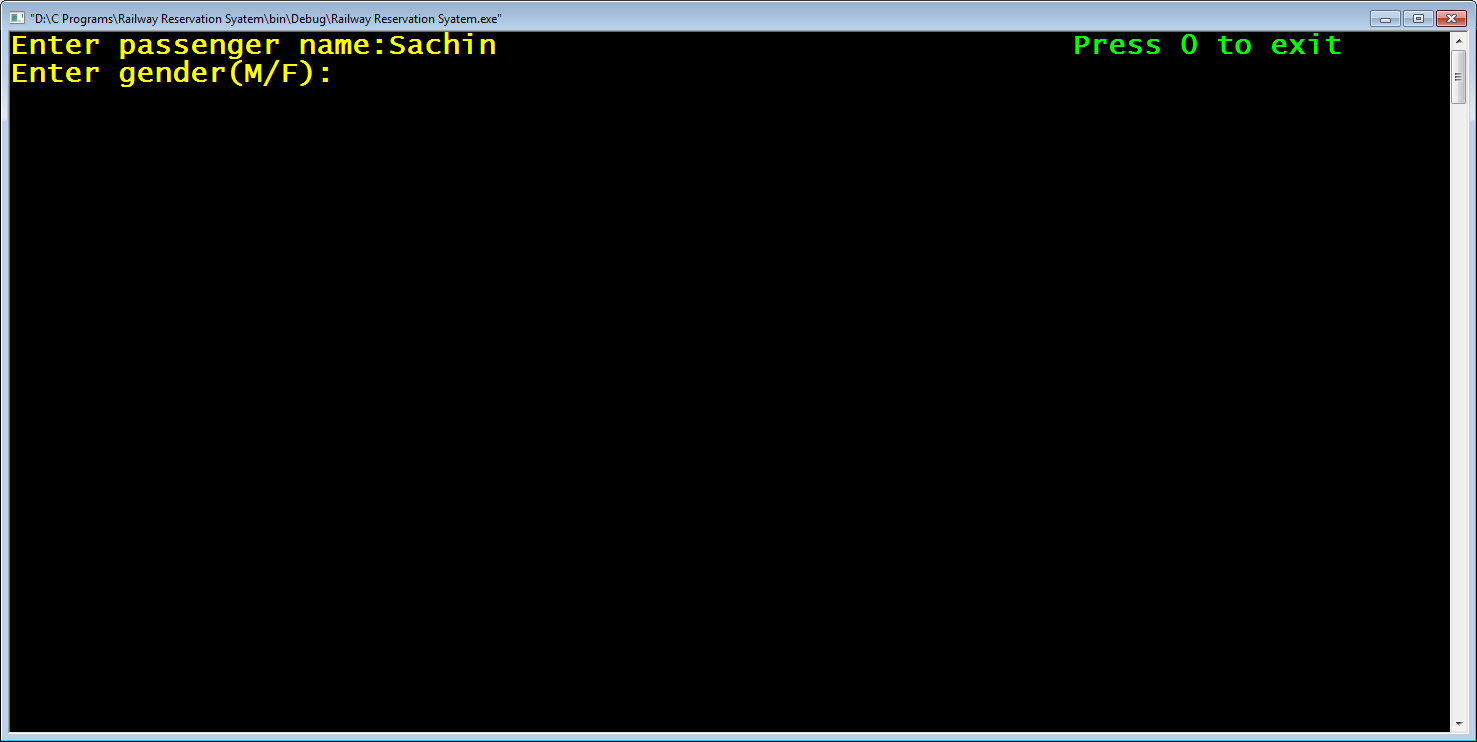
**DESIGNING THE "get\_passenger\_details()" FUNCTION IN THE FILE "rlyres.c"**

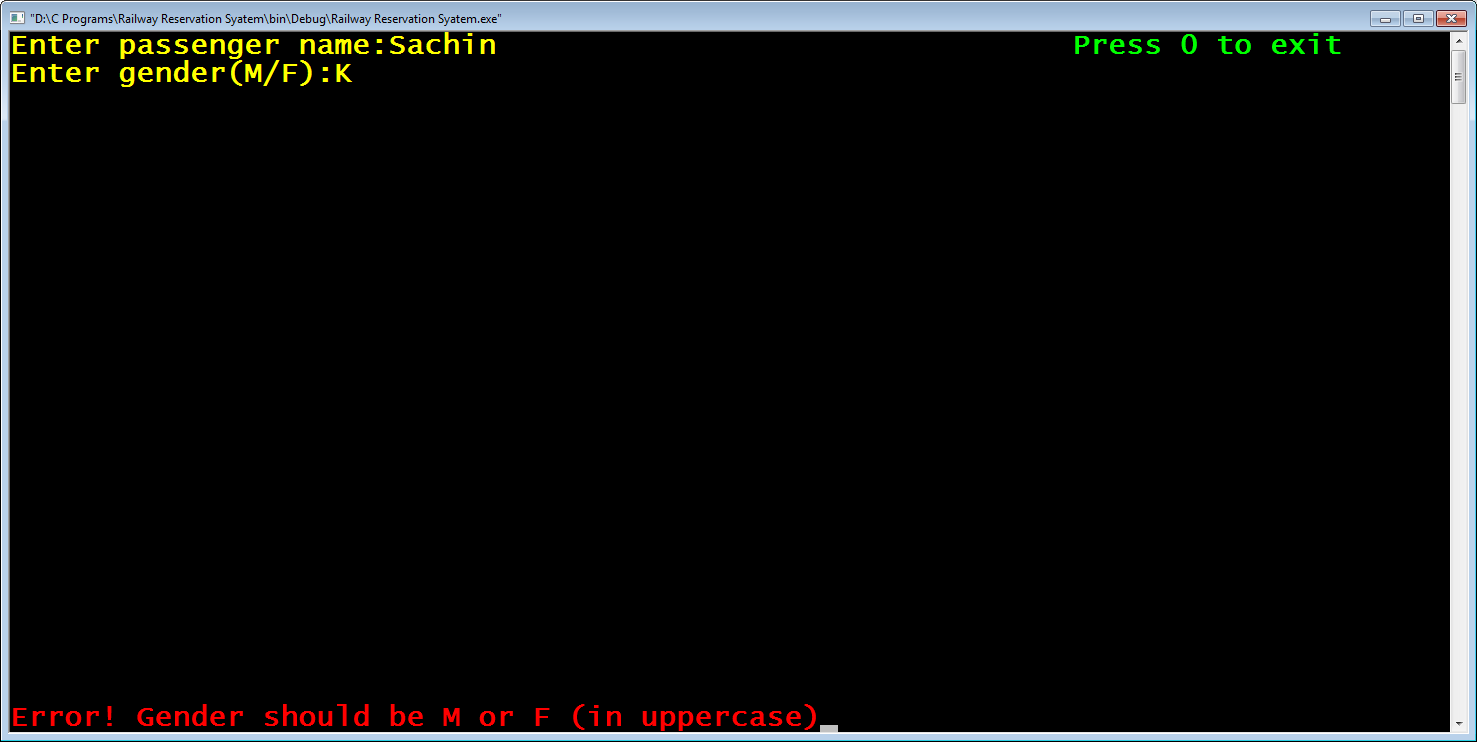
To understand how the function **get\_passenger\_details()** will be designed , we must have a look at some screenshots this function produces:

1. The first screen prompts the user to input his/her name as well as it displays the message "**Press 0 to exit**" at the **top right corner** and this message is displayed continuously.

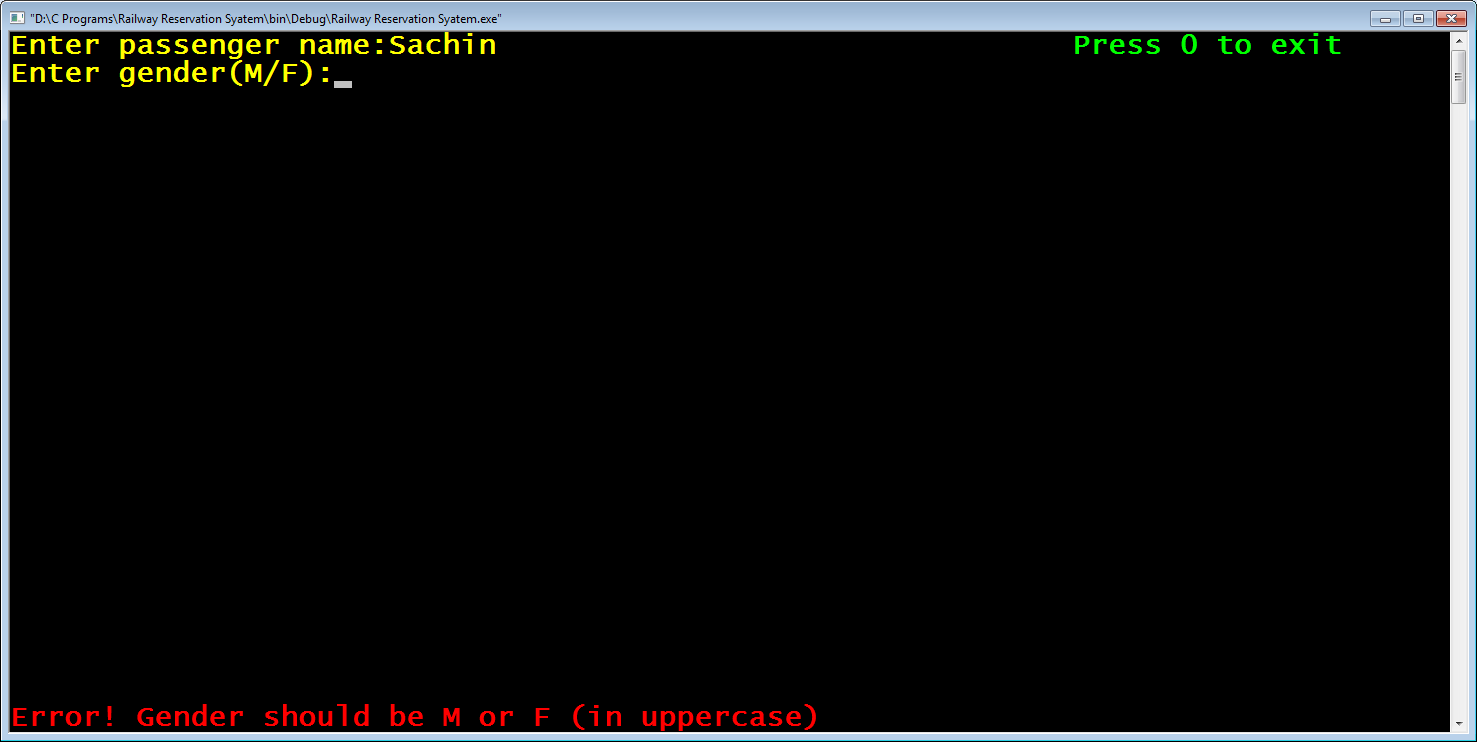


2. After the user inputs his name , the function prompts the user to input his/her gender as **M** or **F**.If any other character is inputted it generates an error message at the **bottom of the screen** as "**Error! Gender should be M or F (in uppercase)**"

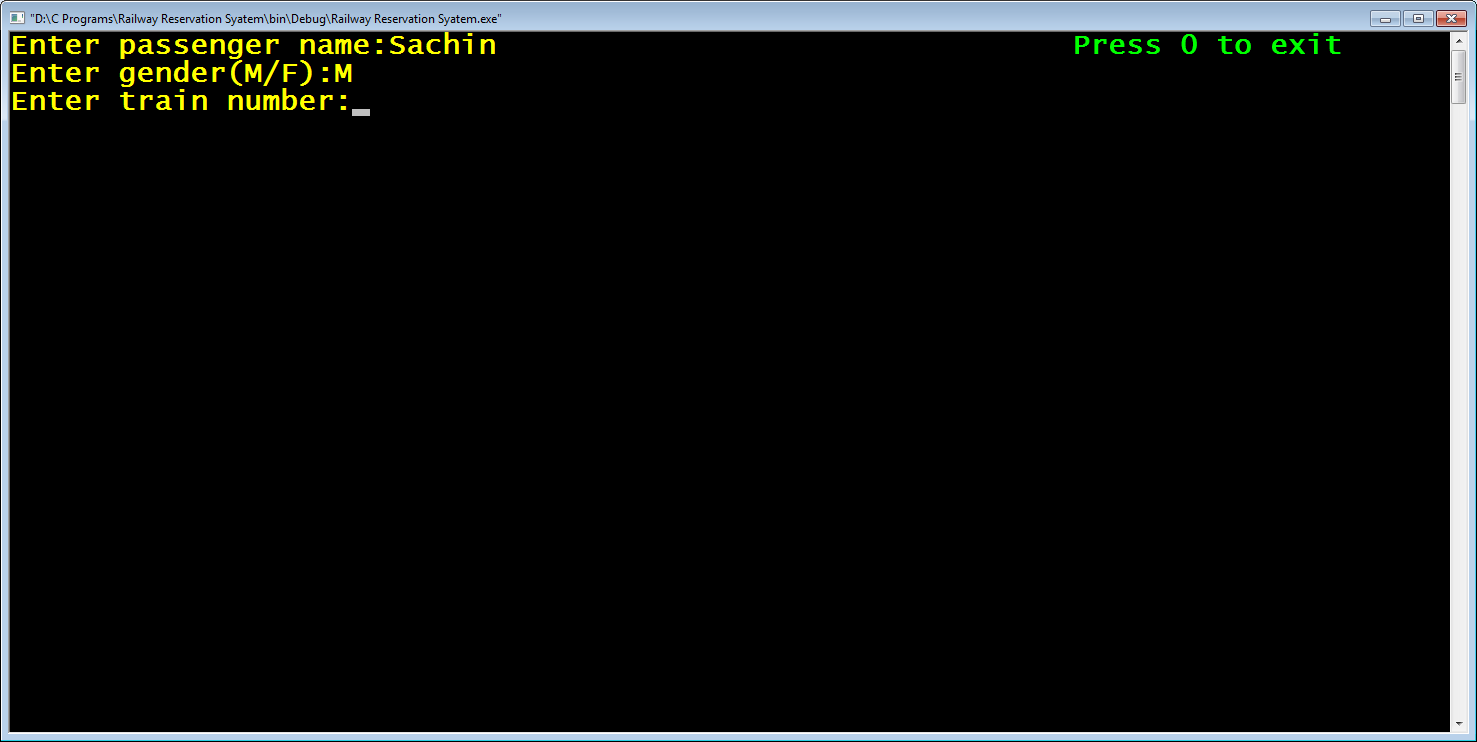




3. The code then again waits for the correct input



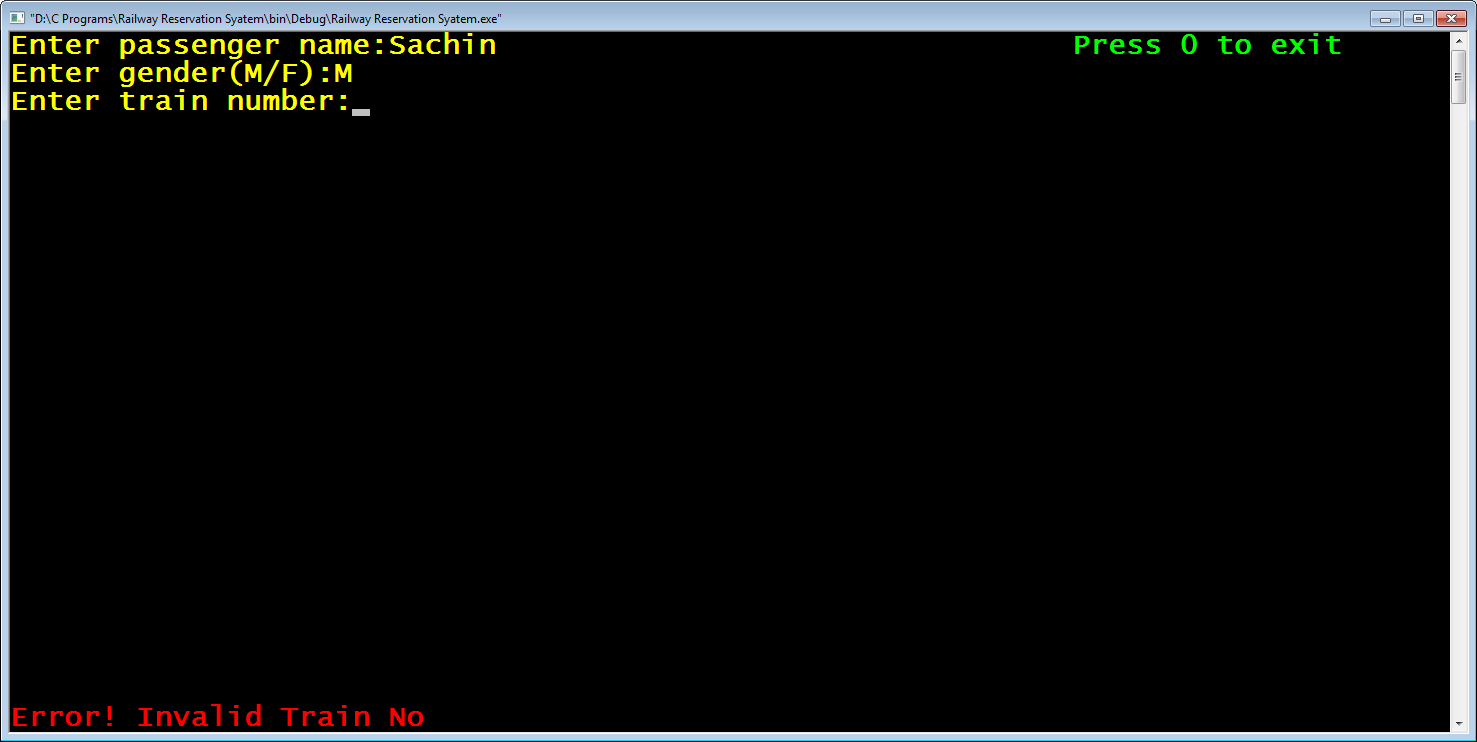
4. After correct gender is given the code , prompts the user to input **train number**



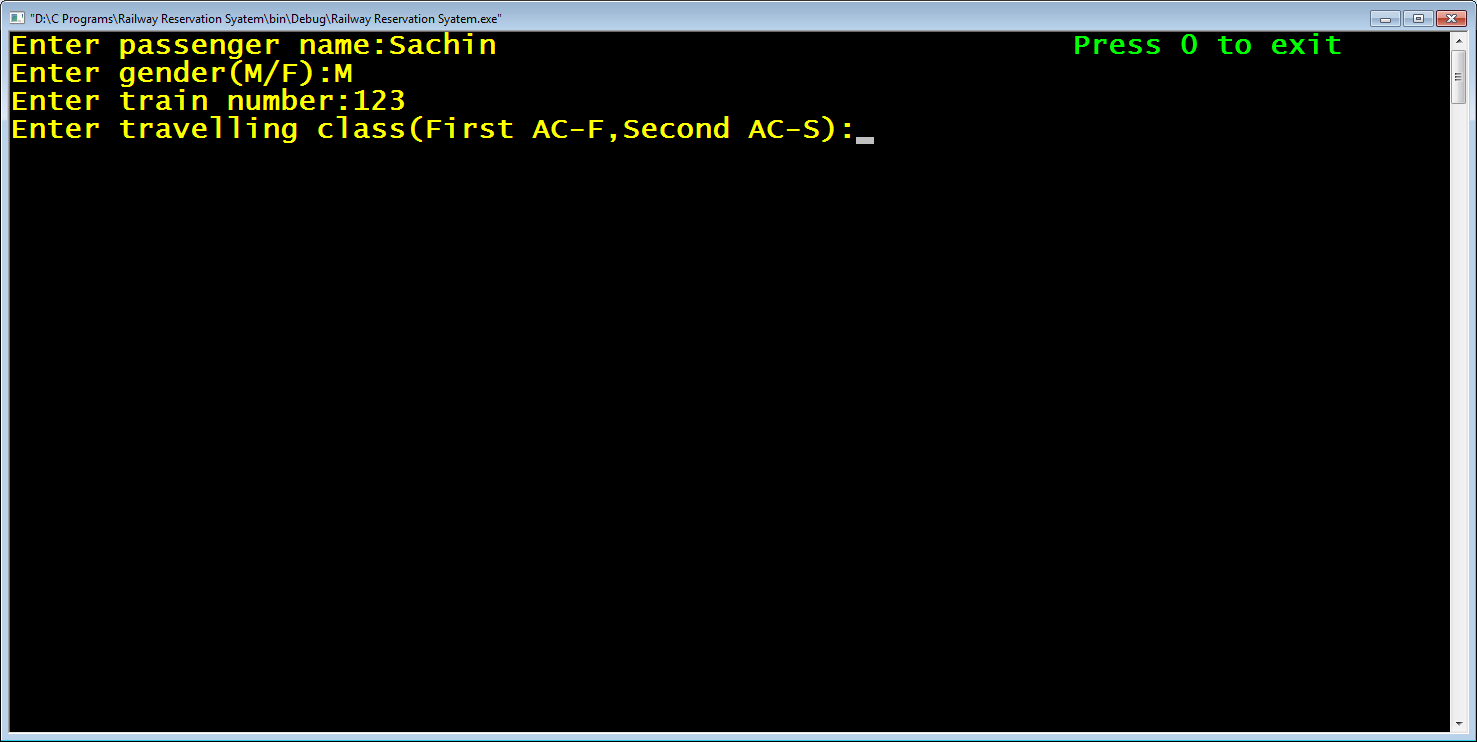
5. It then checks whether train number is valid or not . If it is invalid , it displays the error message as "**Error ! Invalid Train No**".



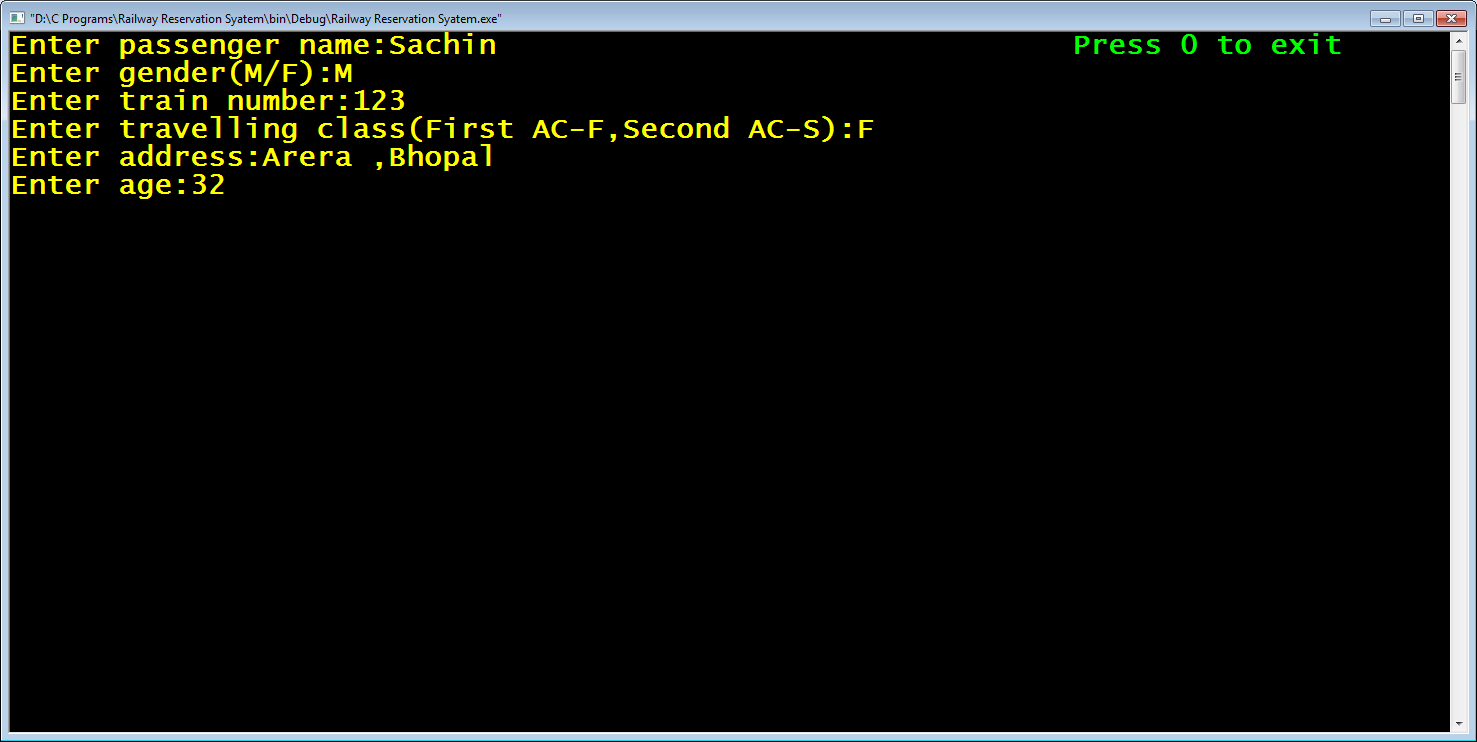
6.The code then again waits for the correct input



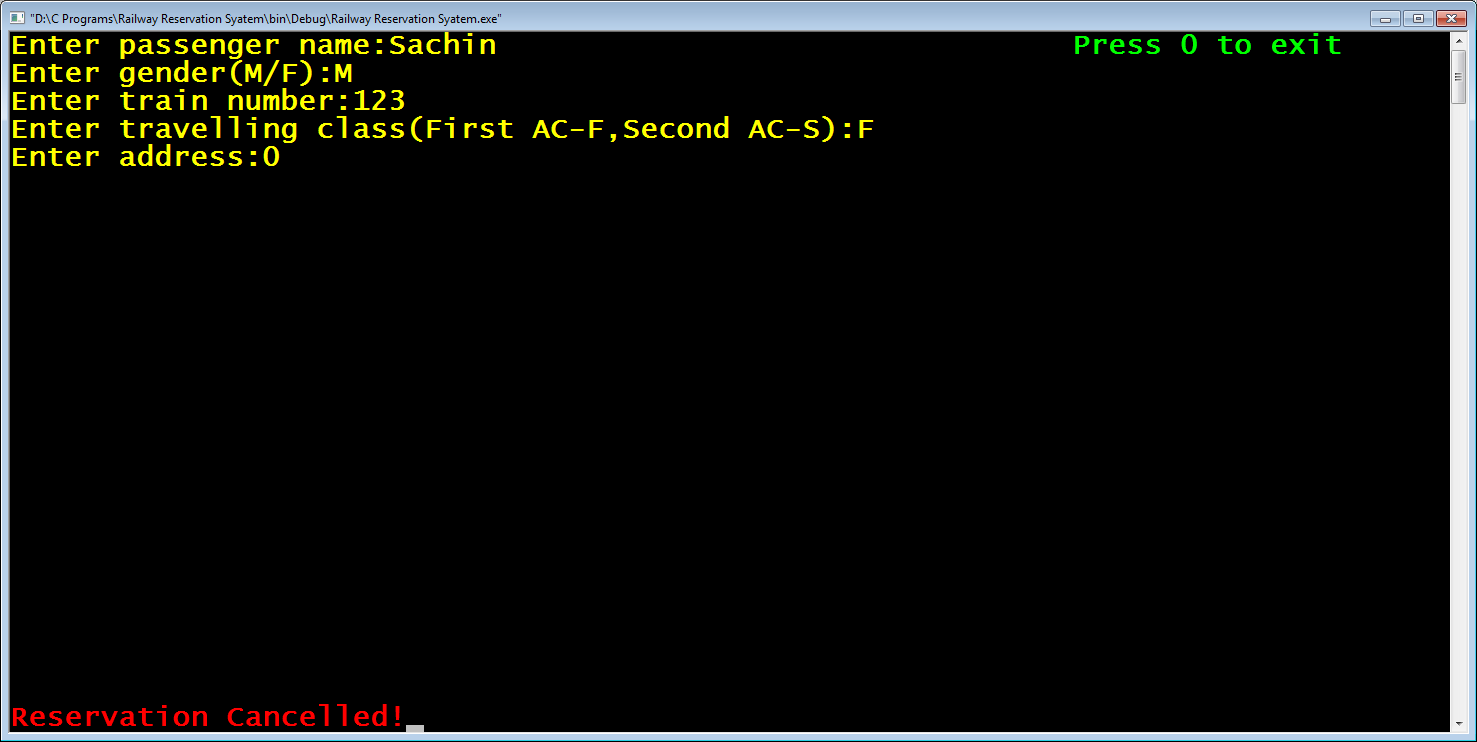
7. If correct train number is given , the code prompts the user to input **travelling class** as **F** or **S**



8. Like this , the code accepts all the inputs with validations.



9. Also , if during these inputs the user types 0 as input , then the function displays the message "**Reservation Cancelled**" and stops accepting further inputs



Following are the stepwise details of the function **get\_passenger\_details()**:

1. **Display the message "Press 0 to exit" at the top right corner**

For this we need to call the functions **gotoxy()** , **textcolor()** and **printf()** with appropriate arguments.

Here is the code for that:

***gotoxy(40,1);***

***textcolor(LIGHTGREEN);***

***printf("Press 0 to exit");***

2. **Accept passenger name from the user and if he presses 0 then display the message "Reservation Cancelled" and return back**

For this following steps are required:

a. Set the cursor at the location **1,1** as well as set the font color to **YELLOW**

Here is the code for it:

***gotoxy(1,1);***

***textcolor(YELLOW);***

***printf("Enter passenger name:");***

b. Declare a **static** variable of type **Passenger**

c. Accept input using the function **fgets( )** ( we are not using the function **gets()** because it suffers from the problem of buffer overflow and it is not at all recommended to be used , while the function **fgets()** has no such problem)

Here is the code for it:

***static Passenger psn;***

***fflush(stdin);***

***fgets(psn.p\_name,20,stdin);***

d. The function **fgets()** has 1 problem. It appends the extra newline character at the end of the input when the user strikes **ENTER** key . So we have to remove the extra newline character that **fgets()** appends at the end.

e. For this we have a function called **strchr()** available in the header file **string.h** . It's prototype is:

**int \* strchr(char \*,char);**

This function accepts a string and a character as argument and returns it's **address** in the given string . So we will search for the **address** of **'\n'** in the string and overwrite **'\0'** at that **address**

Here is the code for it:

***char \*pos;***

***pos=strchr(psn.p\_name,'\n');***

***\*pos='\0';***

f. Check if 0 has been entered . If the user has entered 0 then move to bottom coordinate (1,25) of the screen using **gotoxy()**, set the color to **LIGHTRED** using **textcolor()** ,print the message "**Reservation Cancelled**" , reset the color to **YELLOW** and return **NULL**.

Here is the code for it:

***if(strcmp(psn.p\_name,"0")==0)***

***{***

***textcolor(LIGHTRED);***

***gotoxy(1,25);***

***printf("Reservation Cancelled!");***

***getch();***

***textcolor(YELLOW);***

***return NULL;***

***}***

So for all the above steps the combined code will be:

***Passenger \* get\_passenger\_details()***

***{***

***clrscr();***

***gotoxy(60,1);***

***textcolor(LIGHTGREEN);***

***printf("Press 0 to exit");***

***gotoxy(1,1);***

***textcolor(YELLOW);***

***static Passenger psn;***

***printf("Enter passenger name:");***

***fflush(stdin);***

***fgets(psn.p\_name,20,stdin);***

***char \*pos;***

***pos=strchr(psn.p\_name,'\n');***

***\*pos='\0';***

***if(strcmp(psn.p\_name,"0")==0)***

***{***

***textcolor(LIGHTRED);***

***gotoxy(1,25);***

***printf("Reservation Cancelled!");***

***getch();***

***textcolor(YELLOW);***

***return NULL;***

***}***

3. **Accept gender from the user .If he presses 0 then display the message "Reservation Cancelled" and return back , if he inputs anything other than 'M' or 'F' then display the message** "**Error! Gender should be M or F (in uppercase)**" **and again wait for the input. Repeat this process until the input is either** '**M**' **or** '**F**' **then proceed to accept the next input**

For this following steps are required:

a. Prompt the user to input gender.

b. Run a loop which does the following:

i) accept the input

ii)if the input received is 0 then move to bottom coordinate (1,25) of the screen using **gotoxy()**, **erase the previous message** , set the color to **LIGHTRED** using **textcolor()** ,print the message "**Reservation Cancelled**" , reset the color to **YELLOW** and return **NULL**.

Here is the code for it:

***int valid;***

***printf("Enter gender(M/F):");***

***do***

***{***

***valid=1;***

***fflush(stdin);***

***scanf("%c",&psn.gender);***

***if(psn.gender=='0')***

***{***

***textcolor(LIGHTRED);***

***gotoxy(1,25);***

***printf("\t\t\t\t\t\t\t");***

***gotoxy(1,25);***

***printf("Reservation Cancelled!");***

***getch();***

***textcolor(YELLOW);***

***return NULL;***

***}***

iii)if the input received is neither '**M**' nor '**F**' then move to bottom coordinate (1,25) of the screen using **gotoxy()**, , set the color to **LIGHTRED** using **textcolor()** ,print the message " **Error! Gender should be M or F (in uppercase)**" , move back to coordinate (19,2) , erase the entered character and reset the color to **YELLOW** .

iv) Repeat this until input received is '**M**' or '**F**'

v) After the loop is over , erase the error message from bottom coordinate (1,25)

Here is the code for it:

***if(psn.gender!='M' && psn.gender!='F')***

***{***

***textcolor(LIGHTRED);***

***gotoxy(1,25);***

***printf("Error! Gender should be M or F (in uppercase)");***

***valid=0;***

***getch();***

***gotoxy(19,2);***

***printf(" \b");***

***textcolor(YELLOW);***

***}***

***}while(valid==0);***

***gotoxy(1,25);***

***printf("\t\t\t\t\t\t\t");***

So for all the above steps the combined code will be:

***int valid;***

***printf("Enter gender(M/F):");***

***do***

***{***

***valid=1;***

***fflush(stdin);***

***scanf("%c",&psn.gender);***

***if(psn.gender=='0')***

***{***

***textcolor(LIGHTRED);***

***gotoxy(1,25);***

***printf("\t\t\t\t\t\t\t");***

***gotoxy(1,25);***

***printf("Reservation Cancelled!");***

***getch();***

***textcolor(YELLOW);***

***return NULL;***

***}***

***if(psn.gender!='M' && psn.gender!='F')***

***{***

***textcolor(LIGHTRED);***

***gotoxy(1,25);***

***printf("Error! Gender should be M or F (in uppercase)");***

***valid=0;***

***getch();***

***gotoxy(19,2);***

***printf(" \b");***

***textcolor(YELLOW);***

***}***

***}while(valid==0);***

***gotoxy(1,25);***

***printf("\t\t\t\t\t\t\t");***

4. **Accept train number from the user .If he presses 0 then display the message "Reservation Cancelled" and return back. Check whether the train number is valid or not . If it is invalid then display the message** "**Error! Invalid Train No**" **and again wait for the input. Repeat this process until the user inputs a valid train number and**  **then proceed to accept the next input**

For this following steps are required:

a. Set the cursor to **coordinate (1,3)** and prompt the user to **input train number**.

b. Run a loop which does the following:

i) accept the input

ii)if the input received is 0 then move to bottom coordinate (1,25) of the screen using **gotoxy()**, **erase the previous message** , set the color to **LIGHTRED** using **textcolor()** ,print the message "**Reservation Cancelled**" , reset the color to **YELLOW** and return **NULL**.

iii)Check whether the train umber is valid or not by calling the function **check\_train\_no()** . If it is invalid then move to bottom **coordinate (1,25)** of the screen using **gotoxy()**, , set the color to **LIGHTRED** using **textcolor()** ,print the message " **Error! Invalid Train No**" , move back to **coordinate (20,3)** , erase the entered character and reset the color to **YELLOW** .

iv) Repeat this until a valid train number is entered.

v) After the loop is over , erase the error message from bottom coordinate (1,25)

5. **Accept travelling class from the user .If he presses 0 then display the message "Reservation Cancelled" and return back , if he inputs anything other than 'F' or 'S' then display the message** "**Error! Travelling class should be F or S (in uppercase)**" **and again wait for the input. Repeat this process until the input is either** '**F**' **or** '**S**' **then proceed to accept the next input**

For this following steps are required:

a. Set the cursor to coordinate (1,4) and prompt the user to input travelling class.

b. Run a loop which does the following:

i) accept the input

ii)if the input received is 0 then move to bottom coordinate (1,25) of the screen using **gotoxy()**, **erase the previous message** , set the color to **LIGHTRED** using **textcolor()** ,print the message "**Reservation Cancelled**" , reset the color to **YELLOW** and return **NULL**.

iii)if the input received is neither '**F**' nor '**S**' then move to bottom coordinate (1,25) of the screen using **gotoxy()**, , set the color to **LIGHTRED** using **textcolor()** ,print the message " **Error! Travelling class should be F or S (in uppercase)**" , move back to coordinate (48,4) , erase the entered character and reset the color to **YELLOW** .

iv) Repeat this until input received is '**F**' or '**S**'

v) After the loop is over , erase the error message from bottom coordinate (1,25)

6. **Accept address from the user .Replace newline with '\0' . Check If the user has pressed 0 or not as input. If he has pressed 0 then display the message Reservation Cancelled" and return back. Proceed to accept the next input**

For this following steps are required:

a. Set the cursor to **coordinate (1,5)** and prompt the user to **input address** and accept input.

b. Replace '\n' with '\0'

c. if the input received is 0 then move to bottom coordinate (1,25) of the screen using **gotoxy()**, **erase the previous message** , set the color to **LIGHTRED** using **textcolor()** ,print the message "**Reservation Cancelled**" , reset the color to **YELLOW** and return **NULL**.

7. **Accept age from the user .If he presses 0 then display the message "Reservation Cancelled" and return back , if he inputs a negative value or 0 then display the message** "**Error! Age should be positive**" **and again wait for the input. Repeat this process until the input is positive and** **then proceed to accept the next input**

For this following steps are required:

a. Prompt the user to input age.

b. Run a loop which does the following:

i) accept the input

ii)if the input received is 0 then move to bottom coordinate (1,25) of the screen using **gotoxy()**, **erase the previous message** , set the color to **LIGHTRED** using **textcolor()** ,print the message "**Reservation Cancelled**" , reset the color to **YELLOW** and return **NULL**.

iii)if the input received is 0 or negative then move to bottom coordinate (1,25) of the screen using **gotoxy()**, , set the color to **LIGHTRED** using **textcolor()** ,print the message " **Error! Age should be positive**" , move back to coordinate (11,6) , erase the entered age and reset the color to **YELLOW** .

iv) Repeat this until input received is positive.

v) After the loop is over , return the PASSENGER variable's address to the function main()