**TOPIC:- RAILWAY ENQUIRY**

-----------------------------------------------------------------------------------------------------

PROJECT ON DATA STRUCTURE

-----------------------

GROUP ID :- **G5\_B6**

-----------------------

SUBMITTED BY / GROUP MEMBERS

-------------

SR NO. NAME ENROLLEMNET NUMBER

------ ----- -------------------

1. Prakhar Gupta 13103608

2. Yugal Jain 13103611

3. Monik Bansal 13103596

4. Rishabh Jain 13103594

--------------------------------------------------------------------------------------------------------

**TOPIC USED:-**

® File Handling

® Linked List

® Multi Linked List

® Stack

® Graphics

**PROBLEM STATEMENT**

The program contains the train database which is divided in to four zones namely north,south,east and west.

a)North Zone-Consists of 5 trains and 7 stations.Paths of each train are in the order-

A- 1->4->5->7

B- 1->2->4->5->6

C- 1->2->3

D- 1->5->6->7

E- 4->5->6

b)South Zone- Consists of 7 trains and 9 stations.Paths of each train are in the order-

F- 5->6->7->8->9

G- 1->2->3->8->9

H- 1->2->3->4->5->6

I- 4->5->6->7->8->9

J- 4->6->9

K- 1->4->5->6->9

L- 3->4->7->8->9

c)East Zone- Consists of 3 trains and 4 stations.Paths of each train are in the order-

M- 3->4

N- 1->2->3->4

O- 1->2->3->4

d)West Zone- Consists of 5 trains and 5 stations.Paths of each train are in the order-

P- 1->2->3

Q- 1->3->4->5

R- 2->3->5

S- 1->5

T- 3->4->5

Our program has two departments-

1)Administration Department

2)User’s Enquiry

1. Administration Department-Only the group members have an account and can login into it.They have the power to-
2. Print(to view train details of all trains)
3. Insert(to insert any new train in any zone)
4. Delete(to delete any train in a particular region)
5. User’s Login-Users can search a train by two methods either by source and destination stations or by train name.User’s can also view recent 5 searches made by him or earlier users in that particular field.

CORE FUNCTIONS

® **void gotoxy(int x ,int y);**

To move curser at (x,y) position at output screen

® **void file();**

This is one of our main function . In this we take data from file and then call

create function(passing the array length) which inputs this data in linked list.

File() and create() function are related to each other and therefore some variables are

declared globally to use in these to connect these two functions

® **void create(int);**

This function is used to create a multi-linklist by accepting array length.

® **void welcome();**

This is our *startup page* ,here we have used simple graphics.

®**void home();**

This is our home page where we have 2 types of login department

->Admin department

->User department

® **void admin();**

This is our admin login page ,which needs username and password

So for this we have used structure det , in which,

*username* could be **first name(in capital only) of any of our Group Member** and *password* can be **last 4 digits of his enrollment no*.*** respectively

Ex. Username-PRAKHAR

Password-3608

®  **void traverse();**

This function traverses the link list and print train details zone wise.

® **void common(int);**

This is the most important function in our program.It is used to get the mouse pointer along with linking the pages.

® **void insert();**

This function is used to insert a new train along with its stations.

® **void Delete();**

This function is used to delete an existing train.

® **void push(int);**

This funtion pushes a value in the stack.

® **void pop(int);**

This function is used to pop out a value from the stack.

® **void searching();**

This function is used to search train between the two inputted stations.

® **void detail(struct node \*);**

This page prints Entered Train Detail .

STRUCTURES USED

struct det

{

char name[50] ; (To store name)

char pass[5] ; (To store digits )

};

struct node

{

char tname;

char st;

struct node \*down;

struct node \*next;

};

**ALGORITHM**

**Void main()**

**{**

Graphics initialized()

file();

welcome();

loading();

home();

}

**void file()**

{

Int f=0;

FILE \*fp; //local file pointer

Int q=-1 will be our length of array

char \*g[4] will store file names as such shown below

//We have four zone wise trains

g[0]="north.txt"; //north zone contains 5 trains and 7 stations.

g[1]="south.txt"; //south zone contains 7 trains and 9 stations

g[2]="east.txt"; //east zone contains 3 trains and 4 stations

g[3]="west.txt"; //west zone contains 5 trains and 5 stations

for(h=0;4 times for 4 files;h++)

{

start=p=NULL;

fp=fopen(g[h]…) //File opening

while(ch is equal to getc(fp), till end of file)

{

Increment q by 1;

c[q]=ch; //this c[q] array contains a train name and it’s stations

if(ch is ‘0’)

{

We call fn create(passing q (i.e.,array length));

q=-1;

increment f by 1;

}

}

Close file fp;

}

**void create(int q) //q is the length of our array**

{

int l=0; //l is the length we are incrementing from 0 to q

while(l is smaller than q)

{

temp=allocate node memory by using malloc;

temp->tname=temp->st=temp->next=temp->down=NULL;

if(l is 0)

{

start= end=temp;

temp->tname=c[l]; copying 1st character which is TRAIN NAME to our lnked list

if(p is not equal toNULL)

p->down=temp;

p=start;

}

else

{

temp->st=c[l]; //now TRAIN STATIONS will be copied off

end->next=temp;

end=temp;

Increment l by 1;

}

if(f is 0) f is 0 when each new file is open (see in file function)

{

if(h is 0)

startn=start; //**startn is the starting pointer of our NORTH ZONE linked list**

else if(h is 1)

starts=start; //Similarly starts is the starting pointer of our SOUTH ZONE linked list

else if(h is 2)

starte=start; //same for EAST ZONE

else if(h is 3)

startw=start; //same for WEST ZONE

}

Incement f by 1;

}

Now two fn’s simple graphics is used in these to make good presentation

**Void welcome()**

**{**

**}**

**Void loading()**

**{**

**}**

**void home()**   **//page=0** [This function is also our page 0]

{

Giving heading "WELCOME TO RAILWAY ENQUIRY"

status = initmouse(); checking ,is Mouse working or not

if ( status is 0 )

Mouse support not available.

else

{

showmouseptr();

for(k=0;making 3 rectangles)

{

rectangle(10,120+j,250,150+j);

j=j+70;

}

In rectangle one “ Administrator Login” **[page1]**

Same in 2nd “User's Login” **[page2]**

Same in 3rd “Exit”

S[4] is a array of structure det

Copy "PRAKHAR" to our **S[0]in category name**  [This is for username]

Copy “3608” **to second category in pass** [This is for password]

Similarly do it for rest of 3 group members

then we go for next page so for this **we call fn common(passing 0 in it)**

}

}

**void common(pg= here the page no. of that page comes from which this function is called)**

{

Button=-1, gets address of which button has clicked

X and y are coordinates

getmousepos(&button,&x,&y);

while(!kbhit()) //standard

{

getmousepos(&button,&x,&y);

if(pg is 0 ,i.e,called from page 0 or home page)

{

if( user clicks on admin or 1st block range)

We call admin() fn //page 1

else if(user clicks user or 2nd block range)

We call user() fn //page 2

else if(user clicks on Exit or 3rd block range)

We call Exit() }

}

//single condition for all REturn main menu

else if(user clicks on **Return to main menu button** )

We call back(withpassing 0) fn or home() page

else if(pg is 1,i.e called from admin page)

{

if(user clicks on back button range)

We call back(0); // home() }

else if(pg is 2,i.e. called from user page)

{

if(1st block)

user1(); //page 10

else if(2nd block)

user2(); //page 13

else if(button==1 && x>=540 && x<=635 && y>=330 && y<=360)

home();//home or back(0) same thing }

else if(pg is 3)

{

if(back button)

back(1); //page 1 or admin

else if(block 1)

traverse(); //page 4

else if(2nd block)

insert(); // page 5

else if(3rd page)

Delete(); // page 8

}

else if(pg is 4)

{

if(back button)

back(3); // admin1()

}

else if(pg is 5)

{

if(1st block or NORTH ZONE)

insert1(1); // page 6

else if(2nd block or SOUTH ZONE)

insert1(2); // page 6

else if(3rd page)

insert1(3); // page 6

else if(4th page)

insert1(4); // page 6

else if(back button)

back(3); // page admin1()

}

else if(pg is 6)

{

if(back button)

back(5); // page admin()

}

else if(pg is 8)

{

if(1st block)

Delete1(1); //page 9

else if(2nd )

Delete1(2); //page 9

else if(3rd )

Delete1(3); //page 9

else if(4th )

Delete1(4); //page 9

else if(back button)

back(3); //admin1() or page 3

}

else if(pg is 9) //delete 1

{ if(back button) back(3); // admin1() or page 3}

else if(pg is 10)

{ if(1st )

searching(); //page 11

else if(2nd )

pop(1); //page 12

else if(back button)

back(2); //page 2 or user()

}

else if(pg is 11)

{

if(back button)

back(10); //page 10

}

else if(pg is 12)

{

if(back button)

{

if(u is 1) //u=1 for user1()

back(10);

else if(u is 2) //u=2 for user2()

back(calling page 13);

}

}

else if(pg is 13) //called from user2()

{

if(back button)

back(calling 2); //user(); fn

else if(1st )

zone1(); // page 14

else if(2nd )

pop(2); //page 12

}

else if(pg is 14)

{

if(1st )

detail(startn); // page 15

else if(2nd )

detail(starts); //page 15

else if(3rd )

detail(starte); //page 15

else if(4th )

detail(startw); //page 15

else if(back button)

back(calling page 13); //page 13

}

else if(pg is 15)

if(back button ) back(14); }

} }

**Code Dump**

We had takenTwo functions from net,namely

**® Bmp image functions**

->void loadbitmap(char \*a,int x,int y)

-> int getcol(int col)

This are for getting our train image(which is stord in our file “tra.bmp” ) at startup page when loading fn is called.

**®mouse functions**

->int initmouse()

->void showmouseptr()

->void restrictmouseptr(int x1, int y1, int x2, int y2)

->void getmousepos(int \*button, int \*x, int \*y)

This functions are for showing mouse pointer on the screen and restrict it in our defined size.

Also ,this functions helped to return coordinate of that point where we click from mouse button

Which seems as ,we open a new window or new page when click on the desired button of it.