## horizontal line



Data Structures CE368

09.04.2022

**─**

Your Name

DANISH IRFAN

# Overview

In this Deliverable we show all the menus, create structure and make functions prototype without explanation. We do our best to satisfied you, so we strictly follow all the requirements and barrier in this deliverable.

# Definitions and Explainations

**Definitions:**

* **Doubly Queue**

Double Ended Queue is a type of queue in which insertion and removal of elements can either be performed from the front or the rear. Thus, it does not follow the FIFO rule (First In First Out).

* + - * Next—It is a pointer of DoublyLinkedQueue and used to move forward in queue.
      * Previous—It is a pointer of DoublyLinkedQueue and used to move backward in queue.
      * Front --It is a pointer of DoublyLinkedQueue and used to point the front part of queue.
      * Rear— It is a pointer of DoublyLinkedQueue and used to point the rear part of queue.
* **LinkList**

A linked list is a sequence of links that contain items. Each link contains a connection to another link. The Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

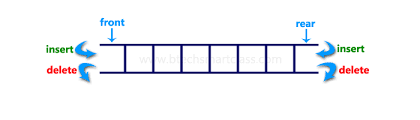
* Element − Each link of a linked list can store data called an element. We make a node in which different types of data can store, like First name, Last name etc.
* Next − Each link of a linked list contains a link to the next link called Next.
* Header – A node which is use to traverse the data.

## **Linked List Representation**

Linked list can be visualized as a chain of nodes, where every node points to the next node.

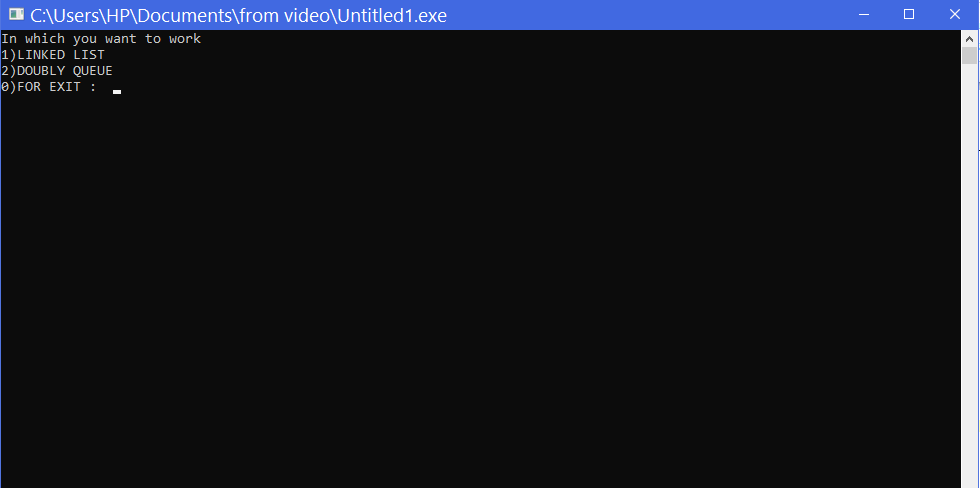


## **Doubly Queue Representation**

Doubly Queue can be visualized as a chain of nodes, where we enque and deque data from both ends (rear and front).**First Menu**

First menu given to the user to select the choice in which he want to work (Linked list or Doubly Queue).

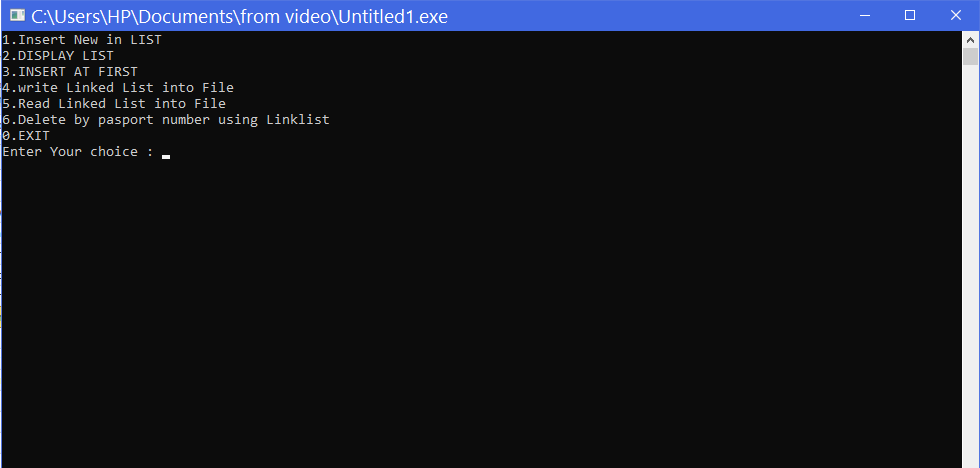
Like



## **Linked List Menu**

After First Menu, if user select “1” then it enter to Linked List Menu.

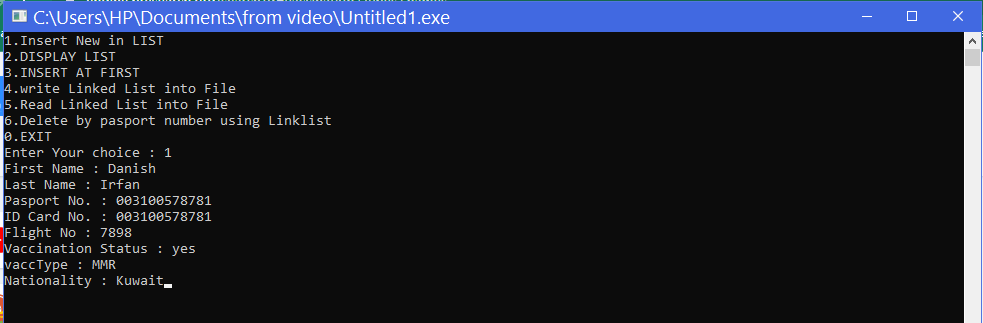
Like



## **Input in Linked List from user**

A linked list is a sequence of data structures, which are connected together via links. Linked List is a sequence of links which contains items. It have nodes like a container and each container contains data items and link of next node. In this section we manually enter the values by user.

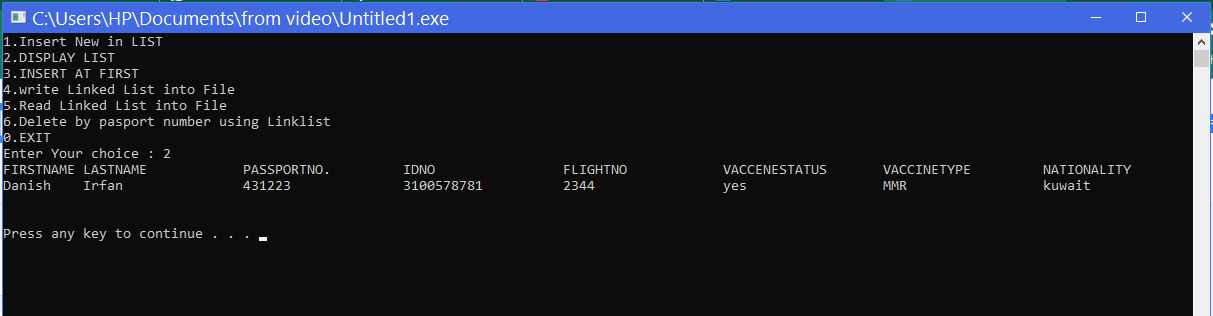
Like



## **Display Linked List**

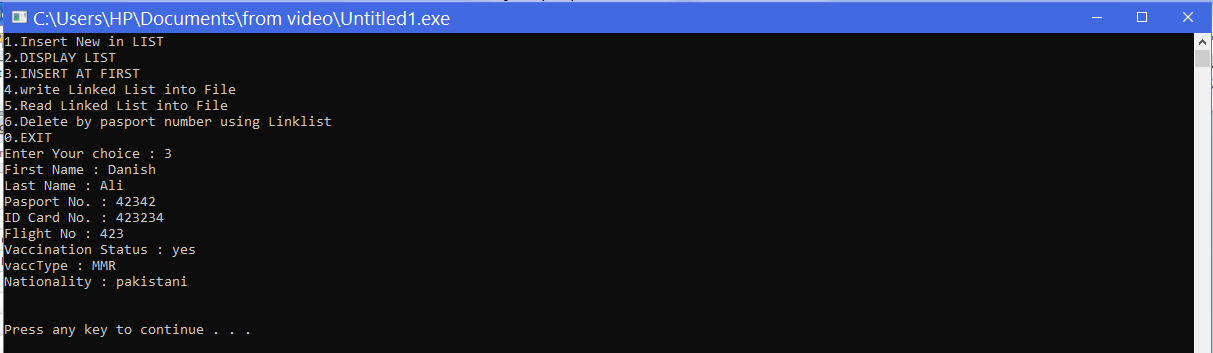
Display all Data of Linked List

Like

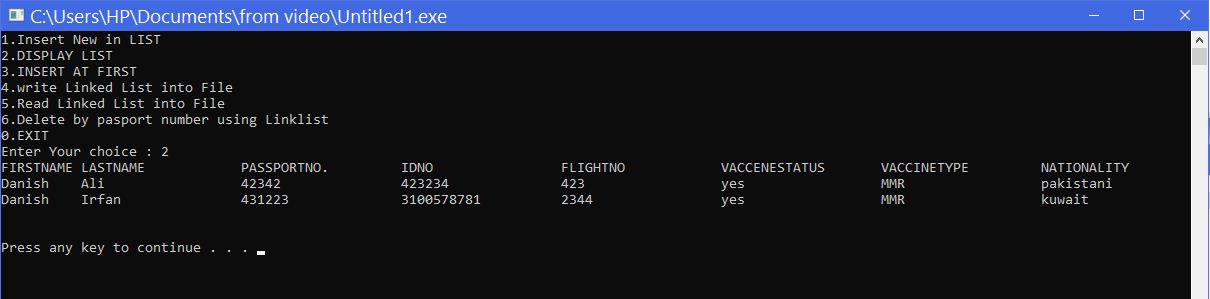


## **Insert at First in Linked List**

Insert entry at the beginning of the Linked list.



After Display



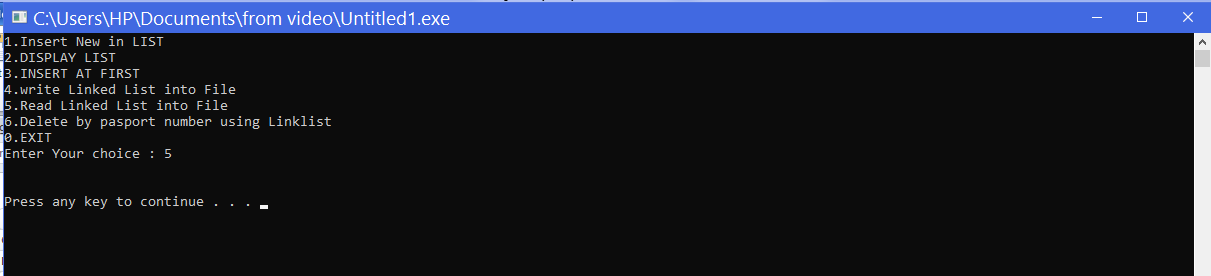
## **Write into the File from Linked List**

This function just writes data into the file.

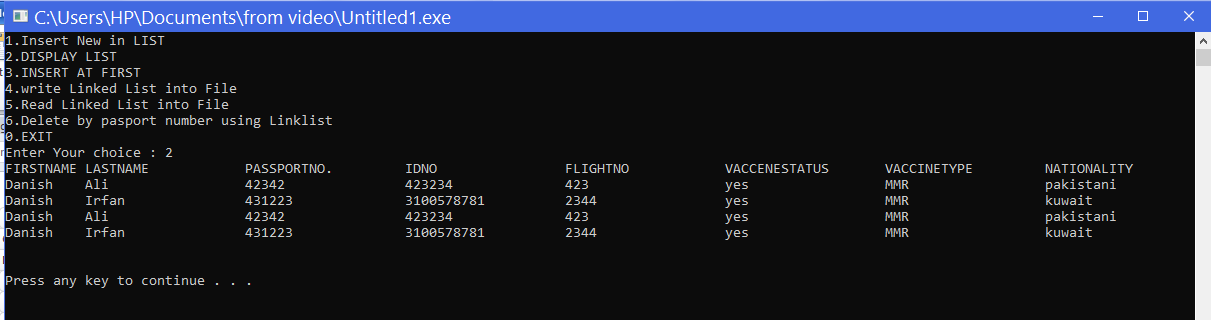


## **Read from the File and store into Linked List**

This function just read data from File and store into Linked List.

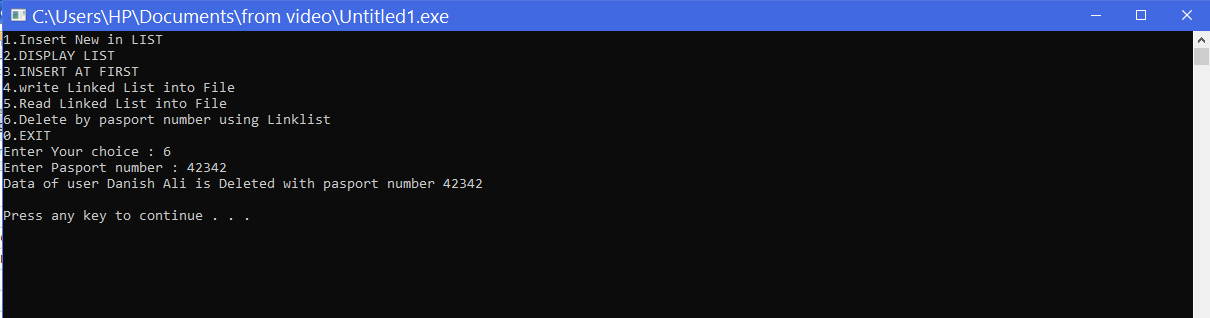


After Read From File our Display Function can show more items. Like…



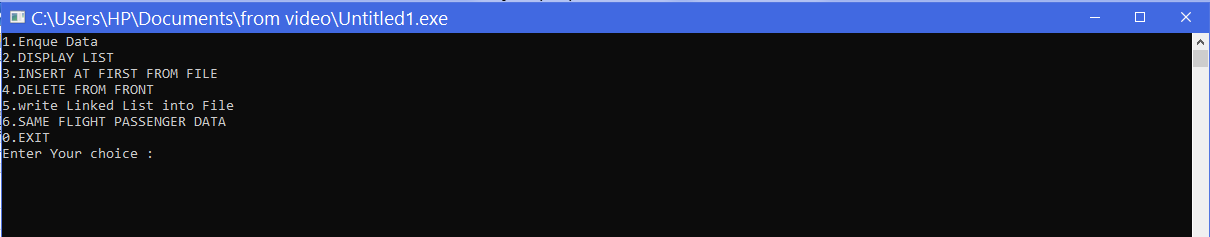
## **Delete by Passport number**

Entry is deleted using passport number.



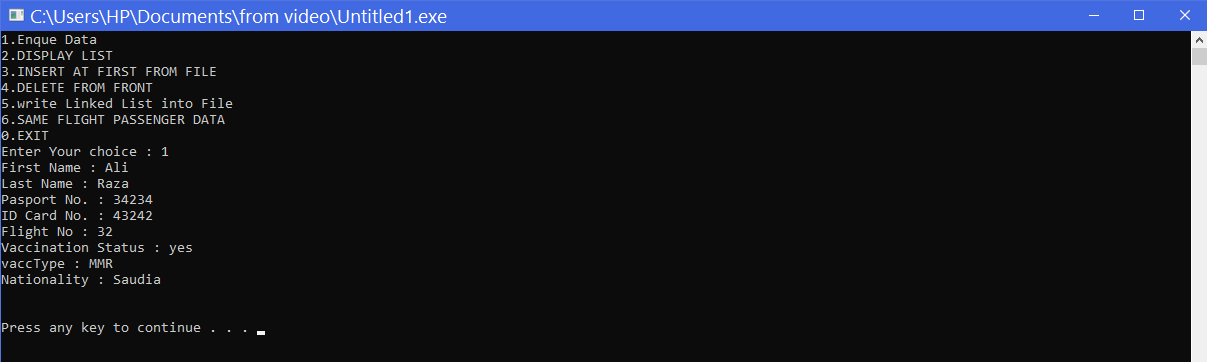
**Menu Doubly Queue**

After pressing “2” in main menu we enter into Doubly Queue Menu.



## **Enque Data into Queue By User**

User enter data into Doubly Queue manually.



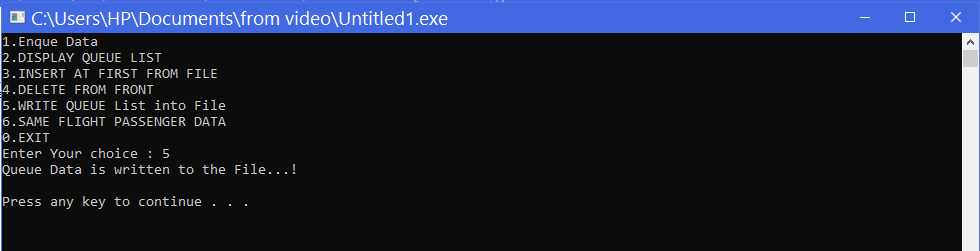
## **Display List of Queue**

Display all entries of Queue.



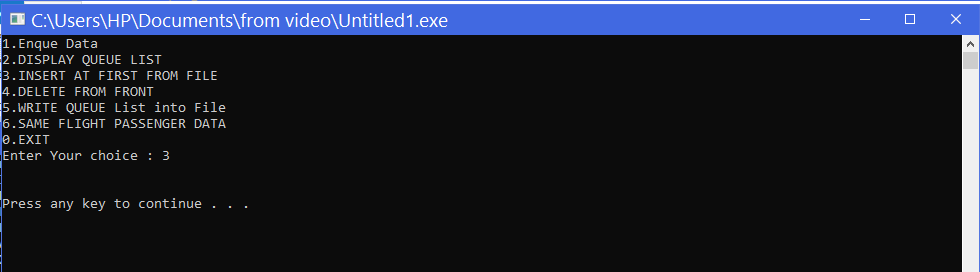
## **Write List of Queue into the File**

Write all Entry of Queue into the file, It is must before read from the file because it store entries into the file for reading purpose.

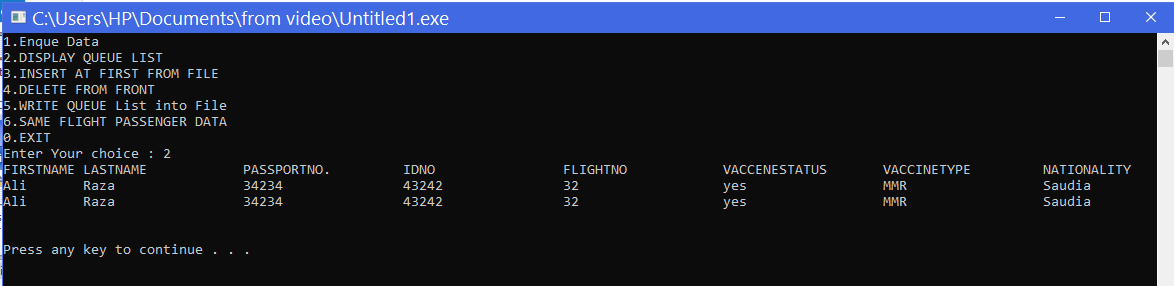


## **Read List from file into the Queue**

It read all data from file and store into queue.

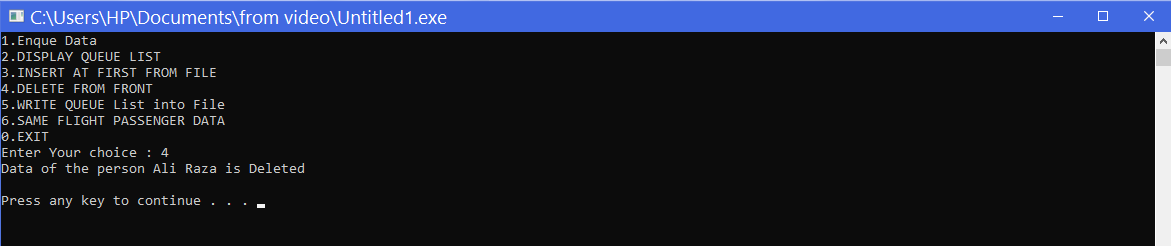


After Reading from File Display of Queue is



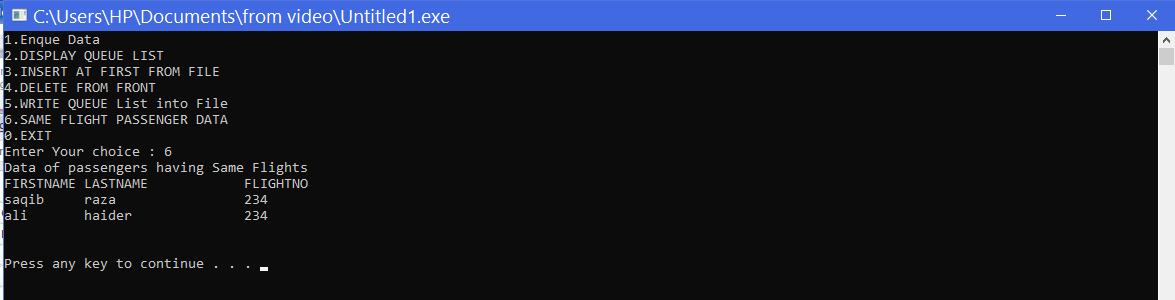
## **DeQueue First Element**

It delete first element of Queue.



## **Same Flight Passenger Data**

It display record of passengers data having same flight.



# Source Code

#include<stdlib.h>

#include<stdio.h>

#include<string.h>

struct Linklist //Linkedlist and their attributes

{

char FirstName[20],lastName[20],pasportNo[12],IDno[15],flightno[20],vaccStatus[5],vaccType[10],nationality[20];

struct Linklist \*next;

};

struct Linklist \* head=NULL;

struct DoublyLinkedQueue{ //Doubly Queue and their attributes

char FirstName[20],lastName[20],pasportNo[12],IDno[15],flightno[20],vaccStatus[5],vaccType[10],nationality[20];

struct DoublyLinkedQueue \* next;

struct DoublyLinkedQueue\* prev;

};

struct DoublyLinkedQueue\* front=NULL;

struct DoublyLinkedQueue\* rear=NULL;

//.........................................................................................................................................................

//..................................................................Queue Functions........................................................................

int isEmpty() // fuction check for Queue is empty or not

{

if(front==NULL && rear==NULL)

{

return 1;

}

return 0;

}

void enqueByUser() // Funtion for use to Enter Value manually and it enter value at the end of Queue

{

struct DoublyLinkedQueue\* new\_node=(struct DoublyLinkedQueue\*)malloc(sizeof(struct DoublyLinkedQueue));

char FirstName[20],lastName[20],pasportNo[12],IDno[15],flightno[20],vaccStatus[5],vaccType[10],nationality[20];

fflush(stdin);

printf("First Name : ");

scanf("%[^\n]s",FirstName);

fflush(stdin);

printf("Last Name : ");

scanf("%[^\n]s",lastName);

fflush(stdin);

printf("Pasport No. : ");

scanf("%[^\n]s",pasportNo);

fflush(stdin);

printf("ID Card No. : ");

scanf("%[^\n]s",IDno);

fflush(stdin);

printf("Flight No : ");

scanf("%[^\n]s",flightno);

fflush(stdin);

printf("Vaccination Status : ");

scanf("%[^\n]s",vaccStatus);

fflush(stdin);

printf("vaccType : ");

scanf("%[^\n]s",vaccType);

fflush(stdin);

printf("Nationality : ");

scanf("%[^\n]s",nationality);

strcpy (new\_node->FirstName, FirstName);

strcpy (new\_node -> lastName, lastName);

strcpy (new\_node -> pasportNo, pasportNo);

strcpy (new\_node -> IDno, IDno);

strcpy (new\_node -> flightno, flightno);

strcpy (new\_node -> vaccStatus, vaccStatus);

strcpy (new\_node -> vaccType, vaccType);

strcpy (new\_node -> nationality, nationality);

new\_node->prev=NULL;

new\_node->next=NULL;

if(isEmpty()) {

front=new\_node;

rear=new\_node;

}

else {

rear->next=new\_node;

new\_node->prev=rear;

rear=new\_node;

}

}

void enqueFromfile() // Get value from file if we already call writeQueueList() function to store value in file

{

struct DoublyLinkedQueue sl;

FILE \*fp = fopen("Project.doc","r");

while(fread(&sl, sizeof(struct DoublyLinkedQueue),1,fp)){

struct DoublyLinkedQueue\* new\_node=(struct DoublyLinkedQueue\*)malloc(sizeof(struct DoublyLinkedQueue));

strcpy (new\_node ->FirstName, sl.FirstName);

strcpy (new\_node -> lastName, sl.lastName);

strcpy (new\_node -> pasportNo, sl.pasportNo);

strcpy (new\_node -> IDno, sl.IDno);

strcpy (new\_node -> flightno, sl.flightno);

strcpy (new\_node -> vaccStatus, sl.vaccStatus);

strcpy (new\_node -> vaccType, sl.vaccType);

strcpy (new\_node -> nationality, sl.nationality);

new\_node->next=NULL;

new\_node->prev=NULL;

if(isEmpty()) {

front=new\_node;

rear=new\_node;

}

else {

new\_node->next=front;

front->prev=new\_node;

front=new\_node;

}

}

fclose(fp);

}

void writeQueueList(){ //use to write value in the file

struct DoublyLinkedQueue \*h=front;

FILE \*fp = fopen("Project.doc","w");

while(h!=NULL){

fwrite(h,sizeof(struct DoublyLinkedQueue),1,fp);

h = h->next;

}

fclose(fp);

printf("Queue Data is written to the File...!");

}

void Deque() // Use for Delete values in the Queue

{

if(isEmpty())

{

printf("The queue is empty");

return;

}

struct DoublyLinkedQueue\* temp=front;

front=temp->next;

if(front==NULL) {

rear=NULL;

}

else {

front->prev=NULL;

}

printf("Data of the person %s %s is Deleted",temp->FirstName ,temp->lastName);

free(temp);

}

void Triverse(struct DoublyLinkedQueue\* front,struct DoublyLinkedQueue\* rear) //Use for Triverse all data in Queue from front side

{

if(isEmpty()){

printf("Queue is Empty");

}

else{

printf("%-10s%-20s%-20s%-20s%-20s%-20s%-20s%s\n","FIRSTNAME","LASTNAME","PASSPORTNO.","IDNO","FLIGHTNO","VACCENESTATUS","VACCINETYPE","NATIONALITY");

while(front!=rear)

{

printf("%-10s%-20s%-20s%-20s%-20s%-20s%-20s%s\n",front->FirstName,front->lastName,front->pasportNo,front->IDno,front->flightno,front->vaccStatus,front->vaccType,front->nationality);

front=front->next;

}

printf("%-10s%-20s%-20s%-20s%-20s%-20s%-20s%s\n",rear->FirstName,rear->lastName,rear->pasportNo,rear->IDno,rear->flightno,rear->vaccStatus,rear->vaccType,rear->nationality);

}

}

void sameFlight(struct DoublyLinkedQueue\* front,struct DoublyLinkedQueue\* rear) //for getting values of same flight passenger

{

if(isEmpty()){

printf("Queue is Empty");

}

else{

printf("Data of passengers having Same Flights\n");

printf("%-10s%-20s%s\n","FIRSTNAME","LASTNAME","FLIGHTNO");

struct DoublyLinkedQueue\* p=front;

struct DoublyLinkedQueue\* q=front;

int i=0;

while(p!=NULL)

{

q=p->next;

while(q!=NULL){

if(strcmp(p->flightno,q->flightno)==0){

i=i+1;

printf("%-10s%-20s%s\n",q->FirstName,q->lastName,q->flightno);

if(i==1){

printf("%-10s%-20s%s\n",p->FirstName,p->lastName,p->flightno);

}

}

q=q->next;

}

p=p->next;

}

if(i==0){

printf("No Data Found Havig same flights\n");

}

}

}

//..............................................................................................................................................

//............................................LINKLIST Functions..........................................................................................

void insertNew(struct Linklist \*h){ //For inserting new Entry by User

char FirstName[20],lastName[20],pasportNo[12],IDno[15],flightno[20],vaccStatus[5],vaccType[10],nationality[20];

fflush(stdin);

printf("First Name : ");

scanf("%[^\n]s",FirstName);

fflush(stdin);

printf("Last Name : ");

scanf("%[^\n]s",lastName);

fflush(stdin);

printf("Pasport No. : ");

scanf("%[^\n]s",pasportNo);

fflush(stdin);

printf("ID Card No. : ");

scanf("%[^\n]s",IDno);

fflush(stdin);

printf("Flight No : ");

scanf("%[^\n]s",flightno);

fflush(stdin);

printf("Vaccination Status : ");

scanf("%[^\n]s",vaccStatus);

fflush(stdin);

printf("vaccType : ");

scanf("%[^\n]s",vaccType);

fflush(stdin);

printf("Nationality : ");

scanf("%[^\n]s",nationality);

if(h==NULL){

head = (struct Linklist\*) malloc(sizeof(struct Linklist));

strcpy (head ->FirstName, FirstName);

strcpy (head -> lastName, lastName);

strcpy (head -> pasportNo, pasportNo);

strcpy (head -> IDno, IDno);

strcpy (head -> flightno, flightno);

strcpy (head -> vaccStatus, vaccStatus);

strcpy (head -> vaccType, vaccType);

strcpy (head -> nationality, nationality);

head -> next = NULL;

}else{

while(h->next != NULL)

h = h-> next;

h -> next= (struct Linklist\*) malloc(sizeof(struct Linklist));

strcpy (head ->next->FirstName, FirstName);

strcpy (head ->next-> lastName, lastName);

strcpy (head ->next-> pasportNo, pasportNo);

strcpy (head ->next-> IDno, IDno);

strcpy (head ->next-> flightno, flightno);

strcpy (head ->next-> vaccStatus, vaccStatus);

strcpy (head ->next-> vaccType, vaccType);

strcpy (head ->next-> nationality, nationality);

h -> next -> next = NULL;

}

}

void viewList(struct Linklist \*h){ //For triversing List

printf("%-10s%-20s%-20s%-20s%-20s%-20s%-20s%s\n","FIRSTNAME","LASTNAME","PASSPORTNO.","IDNO","FLIGHTNO","VACCENESTATUS","VACCINETYPE","NATIONALITY");

while(h!=NULL){

printf("%-10s%-20s%-20s%-20s%-20s%-20s%-20s%s\n",h->FirstName,h->lastName,h->pasportNo,h->IDno,h->flightno,h->vaccStatus,h->vaccType,h->nationality);

h = h -> next;

}

}

void InsertAtBeginning(struct Linklist \*h){ //Inert valus in the Begining of LinkedList

char FirstName[20],lastName[20],pasportNo[12],IDno[15],flightno[20],vaccStatus[5],vaccType[10],nationality[20];

fflush(stdin);

printf("First Name : ");

scanf("%[^\n]s",FirstName);

fflush(stdin);

printf("Last Name : ");

scanf("%[^\n]s",lastName);

fflush(stdin);

printf("Pasport No. : ");

scanf("%[^\n]s",pasportNo);

fflush(stdin);

printf("ID Card No. : ");

scanf("%[^\n]s",IDno);

fflush(stdin);

printf("Flight No : ");

scanf("%[^\n]s",flightno);

fflush(stdin);

printf("Vaccination Status : ");

scanf("%[^\n]s",vaccStatus);

fflush(stdin);

printf("vaccType : ");

scanf("%[^\n]s",vaccType);

fflush(stdin);

printf("Nationality : ");

scanf("%[^\n]s",nationality);

struct Linklist \*temp =(struct Linklist\*) malloc(sizeof(struct Linklist));

strcpy (temp ->FirstName, FirstName);

strcpy (temp -> lastName, lastName);

strcpy (temp -> pasportNo, pasportNo);

strcpy (temp -> IDno, IDno);

strcpy (temp -> flightno, flightno);

strcpy (temp -> vaccStatus, vaccStatus);

strcpy (temp -> vaccType, vaccType);

strcpy (temp -> nationality, nationality);

temp->next = h;

head = temp;

}

void writeLinkedList(struct Linklist \*h){ //For using to write value in file using Linklist

FILE \*fp = fopen("Project.doc","w");

while(h!=NULL){

fwrite(h,sizeof(struct Linklist),1,fp);

h = h->next;

}

fclose(fp);

printf("Linked List Data is written to the File...!");

}

void creatListfromFile(struct Linklist \*h){ //Creating a LinkedList using file but if value already store in file using writeLinkedList() function

struct Linklist sl;

FILE \*fp = fopen("Project.doc","r");

while(fread(&sl, sizeof(struct Linklist),1,fp)){

if(h==NULL){

h = (struct Linklist\*) malloc(sizeof(struct Linklist));

strcpy (h ->FirstName, sl.FirstName);

strcpy (h -> lastName, sl.lastName);

strcpy (h -> pasportNo, sl.pasportNo);

strcpy (h -> IDno, sl.IDno);

strcpy (h -> flightno, sl.flightno);

strcpy (h -> vaccStatus, sl.vaccStatus);

strcpy (h -> vaccType, sl.vaccType);

strcpy (h -> nationality, sl.nationality);

h -> next = NULL;

head = h;

}else{

while(h->next != NULL)

h = h-> next;

h -> next = (struct Linklist\*) malloc(sizeof(struct Linklist));

strcpy (h ->next->FirstName, sl.FirstName);

strcpy (h ->next-> lastName, sl.lastName);

strcpy (h ->next-> pasportNo, sl.pasportNo);

strcpy (h ->next-> IDno, sl.IDno);

strcpy (h ->next-> flightno, sl.flightno);

strcpy (h ->next-> vaccStatus, sl.vaccStatus);

strcpy (h ->next-> vaccType, sl.vaccType);

strcpy (h ->next-> nationality, sl.nationality);

h -> next -> next = NULL;

}

}

fclose(fp);

}

void DeleteByPassport(struct Linklist \*h){ // Delete entry using passport number

char pasp[12];

fflush(stdin);

printf("Enter Pasport number : ");

scanf("%[^\n]s",pasp);

struct Linklist \*current = h;

struct Linklist \*last=h;

while(1){

if(strcmp(current->pasportNo,pasp)==0){

if(current==h){

h=current->next;

} else{

last->next=current->next;

}

printf("Data of user %s %s is Deleted with pasport number %s",current->FirstName,current->lastName,current->pasportNo);

free(current);

break;

}

last=current;

current=current->next;

if(current==NULL){

printf("Data not found along this pasport number %s",pasp);

break;

}

}

}

int main(){

int ccc,ch1,ch2;

do{

printf("In which you want to work\n1)LINKED LIST \n2)DOUBLY QUEUE\n0)FOR EXIT : ");

scanf("%d",&ccc);

switch(ccc){

case 1:

do{

system("cls");

printf("1.Insert New in LIST\n");

printf("2.DISPLAY LIST\n");

printf("3.INSERT AT FIRST\n");

printf("4.write Linked List into File\n");

printf("5.Read Linked List into File\n");

printf("6.Delete by pasport number using Linklist\n");

printf("0.EXIT\n");

printf("Enter Your choice : ");

scanf("%d" ,&ch1);

switch(ch1){

case 1: insertNew(head); break;

case 2: viewList(head); break;

case 3: InsertAtBeginning(head); break;

case 4:writeLinkedList(head);break;

case 5:creatListfromFile(head);break;

case 6: DeleteByPassport(head);break;

case 0: printf("Exiting from Linked List");break;

}

printf("\n\n");

system("pause");

}while(ch1!=0);

break;

case 2:

do{

system("cls");

printf("1.Enque Data\n");

printf("2.DISPLAY QUEUE LIST\n");

printf("3.INSERT AT FIRST FROM FILE\n");

printf("4.DELETE FROM FRONT\n");

printf("5.WRITE QUEUE List into File\n");

printf("6.SAME FLIGHT PASSENGER DATA\n");

printf("0.EXIT\n");

printf("Enter Your choice : ");

scanf("%d" ,&ch2);

switch(ch2){

case 1: enqueByUser(); break;

case 2: Triverse(front,rear); break;

case 3: enqueFromfile(); break;

case 4: Deque();break;

case 5:writeQueueList();break;

case 6:sameFlight(front,rear); break;

case 0: printf("Exiting from Doubly Queue");break;

}

printf("\n\n");

system("pause");

}while(ch2!=0);

break;

case 0:printf("Thanku and Good by\n");break;

}

printf("\n\n");

system("pause");

}while(ccc!=0);

}