Assignment 4

# Name- Akshat Chandrapatle Div – CS – A Roll no. – 45

Write a menu-driven program that maintains a queue of passengers waiting to see a ticket agent. The program user should be able to insert a new passenger at the rear of the queue, display the passenger at the front of the queue, or remove the passenger at the front of the queue. The program will display the number of passengers left in the queue just before it terminates.

Code:  
#include <stdio.h>

#include <stdlib.h>

#include <string.h>

struct QNode {

char data[100];

struct QNode\* next;

};

struct Queue {

struct QNode \*front, \*rear;

int count;

};

struct QNode\* newNode(char k[])

{

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

strcpy(temp->data, k);

temp->next = NULL;

return temp;

}

struct Queue\* createQueue()

{

struct Queue\* q = (struct Queue\*)malloc(sizeof(struct Queue));

q->front = q->rear = NULL;

q->count = 0;

return q;

}

void enQueue(struct Queue\* q, char k[])

{

struct QNode\* temp = newNode(k);

if (q->rear == NULL) {

q->front = q->rear = temp;

} else {

q->rear->next = temp;

q->rear = temp;

}

q->count++;

}

void deQueue(struct Queue\* q)

{

if (q->front == NULL)

return;

struct QNode\* temp = q->front;

q->front = q->front->next;

if (q->front == NULL)

q->rear = NULL;

free(temp);

q->count--;

}

int main()

{

struct Queue\* q= createQueue(1000);

int choice;

printf("\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Ticket Agent Menu\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\n");

printf("\n1.insert a new passenger\n2.remove the passenger\n3.display the passenger at the front\n4.Exit\n");

while(choice != 4)

{

printf("\nEnter your choice : ");

scanf("%d",&choice);

switch(choice)

{

case 1: ;

char name[100];

printf("\nEnter the Full name of passenger: \n");

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

int valid\_name = 1;

for (int i = 0; i < strlen(name); i++) {

if (!isalpha(name[i]) && name[i] != ' ') {

valid\_name = 0;

break;

}

}

if (valid\_name) {

enQueue(q, name);

} else {

printf("Invalid name! Please enter valid name \n");

}

break;

case 2:

if(q->front == NULL){

printf("Queue is empty\n");

break;

}

printf("Passenger %s removed from queue \n",(q->front)->data);

deQueue(q);

break;

case 3:

printf("Queue Front is: %s\n", ((q->front != NULL) ? (q->front)->data : "empty"));

break;

case 4:

printf("Number of passengers in the queue: %d\n", q->count);

exit(0);

break;

default:

printf("\nEnter valid choice \n");

}

}

return 0;

}  
**OUTPUT:**



