**Exercise 5: Write a program to implement QUEUE using arrays that performs following operations (a) INSERT (b) DELETE (c) DISPLAY**

#include <stdio.h>

#define MAX 10

typedef struct

{

int data[MAX];

int f,r;

}QUEUE;

void main()

{

int val,pos,ch;

QUEUE q;

init(&q);

while(1)

{

printf("\nQUEUE OPERATIONS");

printf("\n1. INSERT");

printf("\n2. DELETE");

printf("\n3. PEEK");

printf("\n4. CHANGE");

printf("\n5. DISPLAY");

printf("\n6. EXIT");

printf("\nEnter Option: ");

scanf("%d",&ch);

switch(ch)

{

case 1: printf("\nPlease Enter Value: ");

scanf("%d",&val);

qinsert(&q,val);

break;

case 2: val=qdelete(&q);

if(val!=NULL)

printf("\nDeleted value is: %d\n",val);

break;

case 3: val=qpeek(&q);

if(val!=NULL)

printf("\nFront value is: %d\n",val);

break;

case 4: printf("\nPlease Enter Position: ");

scanf("%d",&pos);

printf("\nPlease Enter Value: ");

scanf("%d",&val);

qchange(&q,pos,val);

break;

case 5: qdisplay(&q);

break;

case 6: exit(0);

break;

default:

printf("\nPlease Enter proper option...");

}

}

}

void init(QUEUE \*q)

{

q->f=-1;

q->r=-1;

printf("QUEUE initialized...");

}

void qinsert(QUEUE \*q,int val)

{

if(q->r==MAX-1)

printf("Queue overflow...\n");

else

{

if(q->f==-1 && q->r==-1)

{

q->f=0;

q->r=0;

}

else

q->r++;

q->data[q->r]=val;

printf("Value %d inserted...\n",val);

}

}

int qdelete(QUEUE \*q)

{

int val=NULL;

if((q->f==-1 && q->r==-1) || (q->f>q->r))

printf("QUEUE underflow...\n");

else

{

val=q->data[q->f];

q->f++;

}

return val;

}

int qpeek(QUEUE \*q)

{

int val=NULL;

if((q->f==-1 && q->r==-1) || (q->f>q->r))

printf("QUEUE underflow...\n");

else

val=q->data[q->f];

return val;

}

void qchange(QUEUE \*q,int pos,int val)

{

int i=q->f+pos-1;

if(i>=q->f && i<=q->r)

q->data[i]=val;

else

printf("Invalid position...\n");

}

void qdisplay(QUEUE \*q)

{

int i;

if((q->f==-1 && q->r==-1) || (q->f>q->r))

printf("QUEUE is empty...\n");

else

{

printf("\n");

for(i=q->f;i<=q->r;i++)

printf("%d ",q->data[i]);

printf("\n");

}

}