**1.Write a menu driven program for circular queue having operations repositioning and removal.**

#include <stdio.h>

#include <stdlib.h>

struct node {

int data;

struct node \*ptr;

} \*front, \*rear, \*shift, \*temp, \*shift1,\*move;

int count = 0;

void create(){

front = rear = NULL;

}

void enque(int d) {

if(rear == NULL) {

rear = (struct node \*)malloc(1\*sizeof(struct node));

rear->data = d;

rear->ptr = NULL;

front=rear;

}

else {

temp = (struct node \*)malloc(1\*sizeof(struct node));

temp->data = d;

temp->ptr = front;

rear->ptr = temp;

rear = temp;

}

count++;

}

void deque() {

shift=front;

if(shift == NULL) {

printf("Empty queue");

}

else {

if(shift->ptr == NULL){

printf("Deleted value: %d ",front->data);

free(front);

front=NULL;

rear=NULL;

}

else {

shift=shift->ptr;

printf("Deleted value: %d ",front->data);

free(front);

front=shift;

}

count--;

}

}

void display() {

int i;

shift=front;

if(count==0) {

printf("EMPTY QUEUE");

}

else {

for(i=0;i<count;i++) {

printf("%d\t",shift->data);

shift=shift->ptr;

}

}

}

int search(int s2) {

int i,g=0;

shift=front;

for(i=0;i<count;i++) {

if(shift->data == s2) {

g=i+1;

break;

}

else {

shift=shift->ptr;

}

}/\*\*< \*/

return (g);

}

void reposition(int d3,int np1) {

int g1,sw,j,k;

g1= search(d3);

if(g1==0) {

printf("Not found!");

}

else {

printf("g1 = %d ",g1);

shift= front;

for( k=0;k<np1-1;k++) {

shift=shift->ptr;

printf("shift= %d ",shift->data);

}

sw=shift->data;

shift1=front;

for( j=0;j<g1-1;j++){

shift1=shift1->ptr;

printf("shift1 = %d \n",shift1->data);

}

shift->data=shift1->data;

shift1->data=sw;

}

}

void removal(int nd,int p1) {

int r1;

temp=(struct node \*)malloc(1\*sizeof(struct node));

temp->data=nd;

move=front;

for( r1=0;r1<p1-2;r1++) {

move=move->ptr;

printf(" move->data %d\t",move->data);

}

temp->ptr=move->ptr;

move->ptr=temp;

count++;

}

int main()

{

printf("Circular queue");

int ch,d1,s1,d2,np,s3,nd1,r12;

create();

while(1) {

printf("\nEnter choice\n 1.Insert 2. Delete 3.Search 4.Repositon 5.Removal 6.Exit ");

scanf("%d",&ch);

switch(ch) {

case 1 :printf("Enter value: ");

scanf("%d",&d1);

enque(d1);

display();

break;

case 2 :deque();

display();

break;

/\*case 3 :printf("Size: %d",count);

break;\*/

case 3: printf("Enter the value to be searched: ");

scanf("%d",&s1);

s3 = search(s1);

if(s3==0) {

printf("Not found");

}

else {

printf("Found at position: %d",s3);

}

break;

case 4: printf("Enter the value to be repositioned: ");

scanf("%d",&d2);

printf("Enter the new position: ");

scanf("%d",&np);

reposition(d2,np);

display();

break;

case 5: printf("Enter the new value: ");

scanf("%d",&nd1);

printf("Enter the position:");

scanf("%d",&r12);

removal(nd1,r12);

display();

break;

case 6: exit(0);

default: printf("Invalid Choice!");

}

}

return 0;

}