3a) WAP to simulate the working of a queue of integers using an array. Provide the following operations: Insert, Delete, Display The program should print appropriate messages for queue empty and queue overflow conditions.

Program:

#include<stdio.h> #define Max 5

int queue[Max]; int front=-1;

int rear=-1;

void insert(int item); void delete();

void display();

void main()

{

int choice, item; while(1)

{

printf("\nMENU\n"); printf("1. Insert\n"); printf("2. Delete\n"); printf("3. Display\n"); printf("4. Exit\n"); printf("Enter your choice: "); scanf("%d", &choice); switch(choice)

{

case 1:

printf("Enter the element to insert: "); scanf("%d", &item);

insert(item); break;

case 2:

delete(); break;

case 3:

display(); break;

case 4:

exit(0);

default:

printf("Invalid choice\n");

}

}

}

void insert(int add\_item)

{

if(rear == Max-1)

{

printf("Queue overflow\n");

}

else

{

if(front == -1)

{

front = 0;

}

rear = rear + 1; queue[rear] = add\_item;

printf("Inserted %d\n", add\_item);

}

}

void delete()

{

if(front == -1 || front > rear)

{

printf("Queue underflow\n"); return;

}

else

{

printf("Deleted item is %d\n", queue[front]); front = front + 1;

}

}

void display()

{

int i;

if(front == -1)

{

printf("Queue is empty\n");

}

else

{

printf("Queue is: ");

for(i = front; i <= rear; i++)

{

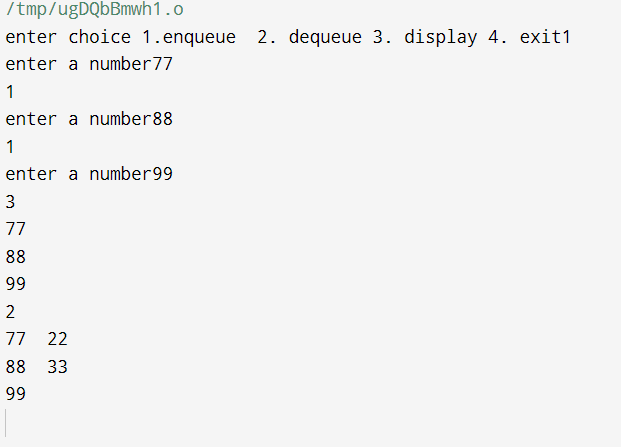
printf("%d ", queue[i]);

}

printf("\n");

}

}



3b ) WAP to simulate the working of a circular queue of integers using an array. Provide the following operations: Insert, Delete & Display The program should print appropriate messages for queue empty and queue overflow conditions.

Program:

#include<stdio.h> #define Max 5

int queue[Max];

int front = -1;

int rear = -1;

void insert(int item); void delete();

void display(); void main() {

int choice, item; while(1) {

printf("\nMENU\n"); printf("1. Insert\n"); printf("2. Delete\n"); printf("3. Display\n"); printf("4. Exit\n"); printf("Enter your choice: "); scanf("%d", &choice);

switch(choice) {

case 1:

printf("Enter the element to insert: "); scanf("%d", &item);

insert(item); break;

case 2:

delete(); break;

case 3:

display(); break;

case 4:

exit(0); default:

printf("Invalid choice\n");

}

}

}

void insert(int item)

{

if ((front == 0 && rear == Max - 1) || (rear == (front - 1) % (Max - 1)))

{

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

}

else if (front == -1)

{

front = rear = 0; queue[rear] = item;

}

else if (rear == Max - 1 && front != 0)

{

rear = 0; queue[rear] = item;

}

else

{

rear++; queue[rear] = item;

}

printf("Inserted %d\n", item);

}

void delete()

{

if (front == -1) {

printf("Queue underflow\n"); return;

}

printf("Deleted item is %d\n", queue[front]); if (front == rear)

{

front = rear = -1;

}

else if (front == Max - 1)

{

front = 0;

}

else

{

front++;

}

}

void display() { int i;

if (front == -1) { printf("Queue is empty\n"); return;

}

printf("Queue is: "); if (rear >= front)

{

for(i = front; i <= rear; i++)

{

printf("%d ", queue[i]);

}

}

else

{

for(i = front; i < Max; i++)

{

printf("%d ", queue[i]);

}

for(i = 0; i <= rear; i++)

{

printf("%d ", queue[i]);

}

}

