**Q. Design a class FIFO queue.**

**🡪** #include <iostream>

#include<conio.h>

#include<stdlib.h>

#define MAX\_SIZE 100

using namespace std;

class Queue {

private:

int item, i;

int arr\_queue[MAX\_SIZE];

int rear;

int front;

public:

Queue() {

rear = 0;

front = 0;

}

void insert() {

if (rear == MAX\_SIZE)

cout << "\n OVERFLOW";

else {

cout << "\nEnter The Value to be Insert : ";

cin>>item;

cout << "\n Position : " << rear + 1 << " , Insert Value : " << item;

arr\_queue[rear++] = item;

}

}

void remove() {

if (front == rear)

cout << "\n UNDERFLOW!";

else {

cout << "\n Position : " << front << " , Remove Value :" << arr\_queue[front];

front++;

}

}

void display() {

cout << "\n Queue Size : " << (rear - front);

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

cout << "\n Position : " << i << " , Value : " << arr\_queue[i];

}

};

int main() {

int choice, exit\_p = 1;

Queue obj;

do {

cout << "\n\n Queue Main Menu";

cout << "\n1.Insert \n2.Remove \n3.Display \nOthers to exit";

cout << "\nEnter Your Choice : ";

cin>>choice;

switch (choice) {

case 1:

obj.insert();

break;

case 2:

obj.remove();

break;

case 3:

obj.display();

break;

default:

exit\_p = 0;

break;

}

} while (exit\_p);

return 0;

}