TermWork - 4

Problem Definition.

Write a C program to simulate working of Messaging System in which a message is placed in a Queue by a message sender, a message is removed from queue by a message reciever, which can also display contents of Queue.

Aim:

The purpose of this TW is to learn the concept of Queues in C language. Basic Operations using queues & implementation of this data structure in solving problems.

Theory:

Like Stack, Queue is a linear structure which follows a particular order in which operations are performed. The order is FIFO. Mainly, the following 4 basic operations are performed on queue:

→ Enqueue : Adds an item to the queue.
→ Dequeue : Removes an item from the queue.
→ Front : Get front item from queue.
→ Rear : Get rear item from queue.

Program:

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#define MAX_SIZE 5
struct msgq {
        char msg[MAX_SIZE][100];
        int rear, front;
};
void initq( struct msgq *q){
        q→front = q→rear= -1;
}
int qfull(struct msgq q){
    return (q.rear == MAX_SIZE -1) ? 1 : 0);
}
int qempty (struct msgq q){
    return ((q.front == -1 && q.rear==-1)||q.front > q.rear) ?1:0);
}
int sender (struct msgq *q, char msg[100]) {
   if (!qfull (*q)){
           if (q→front == -1) q→front =0;
           strcpy (q→msg [++q → rear)], msg);
           return 1;
   }
   printf("\n QUEUE FULL"); EMPTY ");
    return 0;
}
```

```
int receiver (struct msgq *q) {
    if (!qempty (*q)) {
        printf ("Messag = %s", q→msg [q→front]);
        (q→front)++;
        return 1;
    }
  printf ("\n QUEUE EMPTY");
  return 0;
}
~~void displayqueue (struct msgq mq){~~
int main (int argc, char ** argv) {
    struct msgq mq;
    int role, flag;
    char msg [100];
    initq ( &mq);
    while (1) {
    printf ("\n Select your role :\n1- Sender \n 2: Reciever\n 3: Exit ");
    scanf ("%d ",&role);
    if (role == 1) {
        printf ("\n Enter message:");
        if (sender (&mq, msg))
                printf ("\n Message Sent");
        else
                printf ("\n Message is NOT SENT");
  }
  if (role == 2) {
        if (receiver (&mq))
            printf ("\n Message read successfully!");
        else
            printf ("\n No Messages in queue");
}
```

```
    if (role = =3)
          break;
    }
}
```

References:

Books:

* Richard F Gilberg, Behrouz A Forouzan, Data Structures : A Pseudo Code Approach with C, Cengage 2007.

E - Resources:

* https:// geeksforgeeks. org /

Conclusion:

In this TW, I learnt about ~~stacks~~ queues, basic operations of queues & their implementation to solve problems. We also learned basic problem solving techniques & programming paradigms.