

Assignment 3

Subject - Data Structures

U21CS089

Garvit Shah

Q1.

Code -

```
#include <stdio.h>
#include <stdlib.h>
```

```
//Q1
```

```
struct Queue{
    int front;
    int rear;
    char jobs[10][50];
};
```

```
int enqueue(struct Queue* Q){                                // To ADD
    jobs to the Queue
    if(Q->front == -1){
        Q->front +=1;
    }
    if(Q->rear >= 10){
        printf("\nOverflow Error!\n");
    }
    else{
        printf("\nEnter the file to print\n--> ");
        scanf("%s", Q->jobs[Q->rear+1]);
        Q->rear += 1;
    }

    return Q->rear;
}
```

```
int dequeue(struct Queue* Q){                                // To
    DELETE jobs from the Queue
    if(Q->front == -1){
        printf("\nUnderflow Error!\n");
    }
    else{
```

```

        printf("\nFinished: %s\n", Q->jobs[Q->front]);
        Q->front +=1;
    }
    return Q->front;
}

void display(struct Queue* Q){
    if(Q->front == -1){
        printf("No Jobs");
    }
    else{
        printf("\n-----");
        for(int i = Q->front; i<=Q->rear;i++){
            printf("\n|   | Job %d: %s", i, Q->jobs[i]);
        }
        printf("\n-----\n");
    }
}

int main(){
    int a=-1;
    struct Queue* Q;
    Q->front = -1;
    Q->rear = -1;
    while(a){
        printf("\n1. New Job\n2. Finish Current Job\n3. Display
Jobs\n0. Exit\n--> ");
        scanf("%d", &a);
        switch(a){
            case 1:
                enqueue(Q);
                break;
            case 2:
                dequeue(Q);
                break;
            case 3:
                display(Q);
                break;
            case 0:

```

```
        break;
    default:
        break;
    }
}
```

Output -

```
1. New Job
2. Finish Current Job
3. Display Jobs
0. Exit
--> 1
```

```
Enter the file to print
--> File1
```

```
1. New Job
2. Finish Current Job
3. Display Jobs
0. Exit
--> 1
```

```
Enter the file to print
--> Mario
```

```
1. New Job
2. Finish Current Job
3. Display Jobs
0. Exit
--> 1
```

```
Enter the file to print
--> Ash
```

```
1. New Job
2. Finish Current Job
3. Display Jobs
0. Exit
--> 3
```

```
-----  
| | Job 0: File1  
| | Job 1: Mario  
| | Job 2: Ash  
-----
```

```
1. New Job  
2. Finish Current Job  
3. Display Jobs  
0. Exit  
--> 2
```

Finished: File1

```
1. New Job  
2. Finish Current Job  
3. Display Jobs  
0. Exit  
--> 3
```

```
-----  
| | Job 1: Mario  
| | Job 2: Ash  
-----
```

```
1. New Job  
2. Finish Current Job  
3. Display Jobs  
0. Exit  
--> 0
```

Q2

Code -

```
#include <stdio.h>
#include <stdlib.h>
```

```
struct Queue{
    int front;
    int rear;
    int arr[10];
};
```

```
int enqueue(struct Queue* Q, int val)
{
    // To ADD jobs to the Queue
    if(Q->front == -1){
        Q->front +=1;
    }
    if(Q->rear >= 10){
        printf("\nOverflow Error!\n");
    }
    else{
        Q->arr[Q->rear+1] = val;
        Q->rear += 1;
    }

    return Q->rear;
}
```

```
int dequeue(struct Queue* Q){
    // To
    DELETE jobs from the Queue
    int temp;
    if(Q->front == -1){
        printf("\nUnderflow Error!\n");
    }
    else{
        temp = Q->arr[Q->front];
        for(int i=0; i<Q->rear; i++){
            Q->arr[i] = Q->arr[i+1];
        }
        Q->rear -=1;
    }
}
```

```

        if(Q->rear < Q->front){
            Q->front = -1;
            Q->rear = -1;
        }
    }
    return temp;
}

```

```

void delete(struct Queue* Q){
    int temp, i=0;
    while(i != Q->rear){
        temp = dequeue(Q);
        enqueue(Q, temp);
        i+=1;
    }
    dequeue(Q);
}

```

```

void display(struct Queue* Q){                                     // To
DISPLAY
    if(Q->front == -1){
        printf("\nEmpty\n");
    }
    else{
        printf("\n");
        for(int i = Q->front; i<=Q->rear;i++){
            printf("%d ", Q->arr[i]);
        }
        printf("\n");
    }
}

```

```

int main(){
    int a=-1, n, j, val;
    char action1[10] = "PUSH", action2[10] = "POP";
    struct Queue* Q;
    Q = (struct Queue*)malloc(sizeof(struct Queue));
}

```

```

Q->front = -1;
Q->rear = -1;
printf("Enter the no. of initial elements - ");
scanf("%d", &n);
printf("Enter the queue: \n");
for(int i = 0; i < n; i++){
    scanf("%d", &val);
    enqueue(Q, val);
}
display(Q);
for(int i = 1; ; i++){
    int act = 0;
    char action[10];
    scanf("%s", action);
    j = 0;
    while(action[j] != '\0'){
        if(action[j] == action1[j]){
            act = 1;
        }
        if(action[j] == action2[j]){
            act = 2;
        }
        j++;
    }
    if(act == 1){
        scanf("%d", &val);
        enqueue(Q, val);
    }
    else{
        delete(Q);
    }
    display(Q);
}
}

```

Output -

Enter the no. of initial elements - 4

Enter the queue:

1

2

3

4

1 2 3 4

PUSH 5

1 2 3 4 5

POP

1 2 3 4

POP

1 2 3

PO

1 2

POP

1

POP

Empty

PUSH 4

4

PUSH 5

4 5

POP

4

POP

Empty

POP

Underflow Error

