

Lab 3: Queue:

WAP to simulate the working of queue using an array. Provide the following operations

- a) Insert b) Delete c) Display.

```
→ #include <stdio.h>
```

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

```
#define QUE_SIZE 3
```

```
int item, front = 0, rear = -1, q[10];
```

```
void insertrear()
```

```
{
```

```
    if (rear == QUE_SIZE - 1)
```

```
    { printf("Queue overflow\n");
```

```
      return;
```

```
    }
```

```
    rear = rear + 1;
```

```
    q[rear] = item;
```

```
}
```

```
int deletifront()
```

```
{ if (front > rear)
```

```
    { front = 0;
```

```
      rear = -1;
```

```
      return -1;
```

```
    }
```

```
    return q[front++];
```

```
}
```

```
void displayQ()
```

```
{
```



```
int i;
```

```
if (front > rear)
```

```
{ printf("Queue is empty\n");  
  return;  
}
```

```
printf("Contents of queue : ");
```

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

```
{ printf("%d", q[i]);  
}
```

```
printf("\n");
```

```
}
```

```
int main()
```

```
{ int choice;
```

```
  for(;;)
```

```
  { printf("\n 1: insert-rear 2: delete-front  
    3: display 4: exit\n");
```

```
    printf("Enter the choice : ");
```

```
    scanf("%d", &choice);
```

```
    switch(choice)
```

```
    { case 1: printf("Enter the item to be inserted");
```

```
      scanf("%d", &item);
```

```
      insertrear();
```

```
      break;
```



```

case 2: item = deletefront();
        if (item == -1)
            printf("Queue is empty\n");
        else
            printf("item deleted = %.d\n", item);
            break;
case 3: displayQ();
        break;
default: exit(0);
    }
}

```

Output

1: insert-rear 2: delete-front 3: display 4: exit
 Enter the choice: 1
 Enter the item to be inserted: 10.

1: insert-rear 2: delete-front 3: display 4: exit
 Enter the choice: 1
 Enter the item to be inserted: 20.

1: insert-rear 2: delete-front 3: display 4: exit
 Enter the choice: 1
 Enter the item to be inserted: 30

1: insert-rear 2: delete-front 3: display 4: exit
 Enter the choice: 1
 Enter the item to be inserted: 40,
 Queue overflow.

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 3

Contents of the queue: 10 20 30.

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 2.

item deleted = 10.

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 2.

item deleted = 20

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 2

Item deleted = 30.

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 2

Queue is empty

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 3

Queue is empty.

1: insert-rear 2: delete-front 3: display 4: exit

Enter the choice: 4.