

```
1  /*Consider there are 5 Tasks T1, T2, T3, T4 and T5. Task1 takes 50 seconds to execute,
2  similarly Task2 40 sec, Task3 30 sec, Task4 20 sec Task5 10 sec to complete the execution
3  and the condition is each task is given the fixed timestamp of 10 seconds that they can get the
4  CPU for only 10 seconds after which the other task has to get the CPU for their execution.
5  If Task1 has got the CPU for 10 seconds and 40 seconds is left to complete its execution it
6  has to join at the back of Task5. Similarly, for other tasks which has not completed has to
7  join at the back of other tasks for its turn to get the CPU for execution. Identify the suitable
8  data structure and develop a application (C Program) to demonstrate above execution
9  system in the fair fashion.*/
10 #include<stdio.h>
11 #include<stdlib.h>
12 #define size 5
13 int front=0,rear=-1,q[size],count=0;
14 void insertrear(int item)
15 {
16     if(count==size)
17     {
18         printf("queue overflow");
19         return;
20     }
21     rear=(rear+1)%size;
22     q[rear]=item;
23     count++;
24 }
25 int deletefront()
26 {
27     int item;
28     if(count==0) return -1;
29     item = q[front];
30     front=(front+1)%size;
31     count=count-1;
32     return item;
33 }
34 void displayq()
35 {
36     int i,f;
37     if(count==0)
38     {
39         printf("\nqueue is empty");
```

main.c

```
38 {
39     printf("\nqueue is empty");
40     return;
41 }
42 f=front;
43 int c=count;
44 printf("\nProgram in the queue");
45 while(c!=0)
46 {
47     printf("\nExecution time : %d",q[f]);
48     f=((f+1)%size);
49     c--;
50 }
51 }
52 void execution()
53 {
54     if(count==0)
55     {
56         printf("\nqueue is empty");
57         return;
58     }
59     int a,flag=0;
60     while(flag<5)
61     {
62         a=deletefront();
63         if(a!=0)
64         {
65             a-=10;
66             flag=0;
67         }
68         insertrear(a);
69         if(a==0)
70         {
71             flag++;
72         }
73     }
74     printf("\nExecution completed");
75     displayq();
76     front=0,rear=-1;
77 }
```

main.c

```
68     insertrear(a);
69     if(a==0)
70     {
71         flag++;
72     }
73 }
74 printf("\nExecution completed");
75 displayq();
76 front=0,rear=-1;
77 }
78 void main()
79 {
80     int choice,item;
81     for(;;)
82     {
83         printf("\n1.Insert rear\n2.Delete front\n3.Display\n4.Execution\n5.Exit\n");
84         printf("Enter the choice : ");
85         scanf("%d",&choice);
86         switch(choice)
87         {
88             case 1:printf("Enter the execution time of the program :");
89                     scanf("%d",&item);
90                     insertrear(item);
91                     break;
92             case 2:item=deletefront();
93                     if(item== -1)
94                         printf("queue is empty\n");
95                     else
96                         printf("item deleted is %d \n",item);
97                     break;
98             case 3:displayq();
99                     break;
100            case 4:execution();
101                    break;
102            default:exit(0);
103        }
104    }
105 }
106
```



```
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :50
```

```
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :40
```

```
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :30
```

```
1.Insert rear
2.Delete front
3.Display
4.Execution
5.Exit
Enter the choice : 1
Enter the execution time of the program :20
```

```
1.Insert rear
2.Delete front
3.Display
4.Execution
```

4.Execution  
5.Exit  
Enter the choice : 1  
Enter the execution time of the program :10

1.Insert rear  
2.Delete front  
3.Display  
4.Execution  
5.Exit  
Enter the choice : 3

Program in the queue  
Execution time : 50  
Execution time : 40  
Execution time : 30  
Execution time : 20  
Execution time : 10

1.Insert rear  
2.Delete front  
3.Display  
4.Execution  
5.Exit  
Enter the choice : 4

Execution completed  
Program in the queue  
Execution time : 0  
Execution time : 0  
Execution time : 0  
Execution time : 0  
Execution time : 0

1.Insert rear  
2.Delete front  
3.Display  
4.Execution  
5.Exit

```
1  /*Consider the below scenario of linear queue. Design and implement a queue which utilize the
2  free space without shifting the elements of queue.*/
3
4  #include<stdio.h>
5  #include<stdlib.h>
6  #define size 4
7  int item,front=0,rear=-1,q[size],count=0;
8  void insertrear()
9  {
10     if(count==size)
11     {
12         printf("queue overflow\n");
13         return;
14     }
15     rear=(rear+1)%size;
16     q[rear]=item;
17     count++;
18 }
19 int deletefront()
20 {
21     if(count==0) return -1;
22     item=q[front];
23     front=(front+1)%size;
24     count=count-1;
25     return item;
26 }
27 void displayQ()
28 {
29     int i,f;
30     if(count==0)
31     {
32         printf("queue is empty\n");
33         return;
34     }
35     f=front;
36     printf("Contents of queue \n");
37     for(i=1;i<=count;i++)
38     {
39         printf("%d\n",q[f]);
```

```
34 }
35 f=front;
36 printf("Contents of queue \n");
37 for(i=1;i<=count;i++)
38 {
39     printf("%d\n",q[f]);
40     f=(f+1)%size;
41 }
42 }
43 void main()
44 {
45     int choice;
46     for(;;)
47     {
48         printf("\n1:insertrear\n2:deletefront\n3:display\n4:exit\n");
49         printf("enter the choice\n");
50         scanf("%d",&choice);
51         switch(choice)
52         {
53             case 1:printf("enter the item to be inserted\n");
54                 scanf("%d",&item);
55                 insertrear();
56                 break;
57             case 2:item=deletefront();
58                 if(item==-1)
59                     printf("queue is empty\n");
60                 else
61                     printf("item deleted =%d\n",item);
62                 break;
63             case 3:displayQ();
64                 break;
65             case 4:exit(0);
66                 break;
67             default:printf("Invalid choice\n");
68         }
69     }
70 }
71
72
```

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
23

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
45

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
67

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
89
```



```
4:exit
enter the choice
1
enter the item to be inserted
90
queue overflow
```

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =23
```

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
2
item deleted =45
```

```
1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
67
89
```

```
1:insertrear
2:deletefront
3:display
4:exit
```

input

```
4:exit
enter the choice
3
Contents of queue
67
89

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
1
enter the item to be inserted
44

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
3
Contents of queue
67
89
44

1:insertrear
2:deletefront
3:display
4:exit
enter the choice
4
```

```
...Program finished with exit code 0
Press ENTER to exit console.
```