

SIDDHI SINGH

17BIT0028

ITE – 1004

DSA EXERCISE – 2

In a theme park, the Roller-Coaster ride is started only when a good number of riders line up in the counter (say 20 members). When the ride proceeds with these 20 members, a new set of riders will line up in the counter. This keeps continuing. Implement the above scenario of lining up and processing using arrays with Queue ADT.

CODE :

```
#include<stdio.h>

struct queue
{

int arr[20], first, last;

} p;

int
isfull ()
{

if (p.last == 19)

{

return 1;

}
```

```
else  
  
    {  
  
return 0;  
  
}  
  
}
```

```
int  
isempty ()  
{  
  
if (p.last == -1)  
  
    {  
  
return 1;  
  
}  
  
else  
  
    {  
  
return 0;  
  
}
```

```
}
```

```
void
```

```
enqueue ()
```

```
{
```

```
int n;
```

```
printf ("ENTER RIDER'S NUMBER\n");
```

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

```
if (isfull () == 0)
```

```
p.arr[++p.last] = n;
```

```
else
```

```
{
```

```
printf ("Queue is full\n");
```

```
}
```

```
}
```

```
void
```

```
dequeue ()
{

if (isempty () == 0)

    {

printf ("Rider number %c is allowed on to the roller coaster\n",
        p.arr[p.first]);

p.first++;

    }

else

    {

printf ("There are no people in queue\n");

    }

}

int
main ()
{

p.first = 0;
```

```
p.last = -1;
```

```
int choice, d;
```

```
while (choice != 4)
```

```
{
```

```
printf ("Enter 1 to place person in queue\n");
```

```
printf
```

```
    ("Enter 2 to check if queue is full and to allow all riders to enter the roller  
coaster\n");
```

```
printf ("Enter 3 to see all riders in queue\n");
```

```
printf ("Enter 4 to exit\n");
```

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

```
switch (choice)
```

```
{
```

```
case 1:
```

```
{
```

```
    enqueue ();
```

```
break;
```

```
}
```

case 2:

```
{  
    if (isfull () == 1)
```

```
{
```

```
int i;
```

```
for (i = 0; i < 20; i++)
```

```
{
```

```
    dequeue ();
```

```
}
```

```
}
```

```
else
```

```
{
```

```
    printf
```

```
        ("Queue does not consist of enough members and the ride cannot start\n");
```

```
}
```

```
break;
```

```
}
```

case 3:

```
{
```

```
        int i;

    for (i = 0; i <= p.last; i++)

        {
            printf ("%c \n", p.arr[i]);
        }

    }

    printf ("\n");

    case 4:
        break;

    }

    }

    return 0;

}
```

```
1  #include<stdio.h>
2  struct queue
3  {
4
5  int arr[20], first, last;
6
7  } p;
8
9  int
10 isfull ()
11 {
12
13 if (p.last == 19)
14
15 {
16
17 return 1;
18
19 }
20
21 else
22
23 {
24
25 return 0;
26
27 }
28
29 }
```

```
26
27 }
28
29 }
30
31
32 int
33 isempty ()
34 {
35
36 if (p.last == -1)
37
38 {
39
40 return 1;
41
42 }
43
44 else
45
46 {
47
48 return 0;
49
50 }
51
52 }
53
54 }
```



```

50 }
51 }
52 }
53
54
55 void
56 enqueue ()
57 {
58
59 int n;
60
61 printf ("ENTER RIDER'S NUMBER\n");
62
63 scanf ("%d", &n);
64
65 if (isfull () == 0)
66
67 p.arr[++p.last] = n;
68
69 else
70
71 {
72
73 printf ("Queue is full\n");
74
75 }
76
77 }

```

```

78
79
80 void
81 dequeue ()
82 {
83
84 if (isempty () == 0)
85
86 {
87
88 printf ("Rider number %c is allowed on to the roller coaster\n",
89         p.arr[p.first]);
90
91 p.first++;
92
93 }
94
95 else
96
97 {
98
99 printf ("There are no people in queue\n");
100
101 }
102
103 }
104
105

```

```

104
105
106 int
107 main ()
108 {
109
110 p.first = 0;
111
112 p.last = -1;
113
114 int choice, d;
115
116 while (choice != 4)
117 {
118
119
120 printf ("Enter 1 to place person in queue\n");
121
122 printf
123     ("Enter 2 to check if queue is full and to allow all riders to enter the roller coaster\n");
124
125 printf ("Enter 3 to see all riders in queue\n");
126
127 printf ("Enter 4 to exit\n");
128
129 scanf ("%d", &choice);
130
131 switch (choice)
132

```

```

130
131 switch (choice)
132
133 {
134
135 case 1:
136 {
137     enqueue ();
138
139 break;
140 }
141
142 case 2:
143 {
144     if (isfull () == 1)
145
146 {
147
148 int i;
149
150 for (i = 0; i < 20; i++)
151
152 {
153
154 dequeue ();
155
156 }

```

```

154 dequeue ();
155
156 }
157
158 }
159
160     else
161     {
162         printf
163         ("Queue does not consist of enough members and the ride cannot start\n");
164     }
165
166 break;
167 }
168
169 case 3:
170 {
171     int i;
172
173     for (i = 0; i <= p.last; i++)
174     {
175         {
176             printf ("%c \n", p.arr[i]);
177         }
178     }
179
180     printf ("\n");
181

```

```

164     }
165
166 break;
167 }
168
169 case 3:
170 {
171     int i;
172
173     for (i = 0; i <= p.last; i++)
174     {
175         {
176             printf ("%c \n", p.arr[i]);
177         }
178     }
179
180     printf ("\n");
181
182 case 4:
183     break;
184 }
185 }
186
187 }
188
189 return 0;
190
191 }
192

```

OUTPUT

```
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
1
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
2
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
3
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
4
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
```

```
Enter 4 to exit
1
ENTER RIDER'S NUMBER
4
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
5
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
6
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
7
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
```

```
7
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
8
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
9
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
10
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
11
Enter 1 to place person in queue
```

```
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
12
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
13
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
14
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
15
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
```

```
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
16
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
17
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
18
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
19
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
```

```
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
17
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
18
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
19
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
20
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
```

```
Enter 1 to place person in queue
Enter 2 to check if queue is full and to allow all riders to enter the roller coaster
Enter 3 to see all riders in queue
Enter 4 to exit
1
ENTER RIDER'S NUMBER
21
Queue is full
```