

```

1  #include <stdio.h>
2  #include <stdlib.h>
3
4  typedef int QueueElementType;
5
6  typedef struct QueueNode *QueuePointer;
7
8  typedef struct QueueNode
9  {
10     QueueElementType Data;
11     QueuePointer Next;
12 } QueueNode;
13
14 typedef struct
15 {
16     QueuePointer Front;
17     QueuePointer Rear;
18 } QueueType;
19
20 typedef enum {
21     FALSE, TRUE
22 } boolean;
23
24
25 void CreateQ(QueueType *Queue);
26 boolean EmptyQ(QueueType Queue);
27 void AddQ(QueueType *Queue, QueueElementType Item);
28 void RemoveQ(QueueType *Queue, QueueElementType *Item);
29
30 /*???????? ?????? ????? ?????????? AddQ ????? ?? ?????? ?????????? ? ?????.* /
31 int main()
32 {
33
34     /*?????? ?????????????.* /
35     QueueType AQueue;
36     QueuePointer TempPtr;
37     int i,N,M;
38
39     /*????????????? ??????.* /
40     CreateQ(&AQueue);
41
42     /*????????? ?????????????? ??? ?????????? ?????????? ??????????????????.* /
43     do{
44         printf("DWSE TO PLITHOS TWN FYLAKISMENWN:");
45         scanf("%d",&N);
46
47         if(N < 1 )
48             printf("Try again,the number must me >=1.\n");
49
50     }while(N < 1);
51
52     /*????????? ?????? ?????? ?????????? ?????? ??? ?? ??????? ??? ??????????????????.* /
53     for(i=1; i <= N; i++)
54         AddQ(&AQueue,i);
55
56     /*????????? ?????????????? ??? ?????????? ?????????? ??????????????.* /
57     do{
58         printf("DWSE TON ARITHMO TOU KYKLOU EKTELESIS:");
59         scanf("%d",&M);
60
61         if(M < 1 )
62             printf("Try again,the number must me >=1.\n");
63
64     }while(M < 1);
65
66

```

```

67  /*???? ?????????? ?????? ?? ?????? ??? ???? ??????????????*/
68  while(AQueue.Front != AQueue.Front->Next)
69  {
70
71      for(i=1; i < M; i++)
72      {
73          printf("%d, ",AQueue.Front->Data);
74
75          /*?? ????? ? ?????????????? ????????????? (???????? ?? ??? ?????? ??? ?????? ??????????)
76          ??? ?????? ?? Next ??? ?? ?????? ??? ?????????? ?????????,????? ?? ?????? ??
77          "?????????????".
78          ?? ?? ?????? ?? ?????????????? ,????? ?????????? ??? ??????????*/
79
80          if(i == M-1)
81          {
82              TempPtr = AQueue.Front;
83              AQueue.Front = AQueue.Front->Next;
84              TempPtr->Next = AQueue.Front->Next;
85          }
86          else
87          {
88              AQueue.Front = AQueue.Front->Next;
89          }
90      }
91
92      /*????????? ??? ?????????? ??????????????, ? ?????? ?? ?????????? ??????*/
93      printf("%d, ",AQueue.Front->Data);
94      printf("EXECUTION %d\n",AQueue.Front->Data);
95
96      /*????????? ??? ?????? ??????????*/
97      AQueue.Front = AQueue.Front->Next;
98
99  }
100
101  /*????????? ??? ?????????????? ??? ??????*/
102  printf("SURVIVAL %d",AQueue.Front->Data);
103
104  return 0;
105 }
106
107
108 void CreateQ(QueueType *Queue)
109 {
110     Queue->Front = NULL;
111     Queue->Rear = NULL;
112 }
113
114 boolean EmptyQ(QueueType Queue)
115 {
116     return (Queue.Front==NULL);
117 }
118
119 void AddQ(QueueType *Queue, QueueElementType Item)
120 {
121     QueuePointer TempPtr;
122
123     TempPtr= (QueuePointer)malloc(sizeof(struct QueueNode));
124     TempPtr->Data = Item;
125     /*?? NEXT ??? ?????????? ?????? ?? ?????? ??? ?????? ??????*/
126     TempPtr->Next = Queue->Front;
127     if (Queue->Front==NULL)
128     {
129         Queue->Front=TempPtr;
130     }
131     else
132         Queue->Rear->Next = TempPtr;

```

```
133     Queue->Rear=TempPtr;
134 }
135
136 void RemoveQ(QueueType *Queue, QueueElementType *Item)
137 {
138     QueuePointer TempPtr;
139
140     if (EmptyQ(*Queue)) {
141         printf("EMPTY Queue\n");
142     }
143     else
144     {
145         TempPtr = Queue->Front;
146         *Item=TempPtr->Data;
147         Queue->Front = Queue->Front->Next;
148         free(TempPtr);
149         if (Queue->Front==NULL) Queue->Rear=NULL;
150     }
151 }
152
153
154
```