


Queue

 C:\WINDOWS\system32\cmd.exe

```
Microsoft Windows [Version 10.0.17134.345]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Microsoft>cd C:\Borland\BCC55\Bin

C:\Borland\BCC55\Bin>notepad queue.cpp

C:\Borland\BCC55\Bin>bcc32 queue.cpp
Borland C++ 5.5.1 for Win32 Copyright (c) 1993, 2000 Borland
queue.cpp:
Turbo Incremental Link 5.00 Copyright (c) 1997, 2000 Borland

C:\Borland\BCC55\Bin>queue
Queue 1:
Front=>  1  4  7 <=Rear
Queue 2:
Front=>  2  5  8 <=Rear
Queue 3:
Front=>  3  6  9 <=Rear

C:\Borland\BCC55\Bin>
```

Code ที่ผมแก้

queue.cpp *ปรับแล้ว (เพื่อแก้ไขในผู้คำนวณได้)

```
1  #include <stdio.h>
2  #include <stdlib.h>
3  typedef struct node
4  {
5      void * dataPtr;
6      struct node * next;
7  } QUEUE_NODE;
8  typedef struct
9  {
10     QUEUE_NODE * front;
11     QUEUE_NODE * rear;
12     int count;
13 } QUEUE;
14 QUEUE * CreateQueue(void);
15 bool enqueue (QUEUE * queue, void * itemPtr);
16 void printQueue(QUEUE * stack);
17 int main (void)
18 {
19     QUEUE * queue1;
20     QUEUE * queue2;
21     QUEUE * queue3;
22     int * numPtr;
23     int * itemPtr;
24     queue1 = CreateQueue();
25     queue2 = createQueue();
26     queue3 = createQueue();
27     int i = 1;
28     numPtr = (int *) malloc (sizeof(i));
29     *numPtr = i;
30     enqueue(queue1, numPtr);
31     i = 4;
32     numPtr = (int *) malloc(sizeof(i));
33     *numPtr = i;
34     enqueue(queue1, numPtr);
35     i = 7;
36     numPtr = (int *) malloc(sizeof(i));
```

```

37.  *numPtr = i;
38.  enqueue(queue1, numPtr);
39.  i = 2;
40.  numPtr = (int*) malloc(sizeof(i));
41.  *numPtr = i;
42.  enqueue(queue2, numPtr);
43.  i = 5;
44.  numPtr = (int*) malloc(sizeof(i));
45.  *numPtr = i;
46.  enqueue(queue2, numPtr);
47.  i = 8;
48.  numPtr = (int*) malloc(sizeof(i));
49.  *numPtr = i;
50.  enqueue(queue2,
51.  i = 5;
52.  numPtr = (int*) malloc(sizeof(i));
53.  *numPtr = i;
54.  enqueue(queue3, numPtr);
55.  i = 6;
56.  numPtr = (int*) malloc(sizeof(i));
57.  *numPtr = i;
58.  enqueue(queue3, numPtr);
59.  i = 9;
60.  numPtr = (int*) malloc(sizeof(i));
61.  *numPtr = i;
62.  enqueue(queue3, numPtr);
63.  printf("Queue 1:\n");
64.  printQueue(queue1);
65.  printf("Queue 2:\n");
66.  printQueue(queue2);
67.  printf("Queue 3:\n");
68.  printQueue(queue3);
69.  return 0;
70.  }
71.  QUEUE *createQueue(void)
72.  {
73.      QUEUE *queue;
74.      queue = (QUEUE*) malloc(sizeof(QUEUE));
75.      if(queue)
76.          {

```

```
112 void printQueue(Queue * queue)
113 {
114     Queue_NODE * node = queue->front;
115     printf("Front=>");
116     while (node)
117     {
118         printf("%d", *(int*) node->dataPtr);
119         node = node->next;
120     }
121     printf("<=Rear\n");
122     return;
123 }
```