queue.cpp ที่ใช้รันจริง

```
#include <stdio.h>
#include <stdlib.h>
typedef struct node
{
void * dataPtr;
struct node* next;
} QUEUE_NODE;
typedef struct
{
QUEUE_NODE*front;
QUEUE_NODE* rear;
int count;
} QUEUE;
QUEUE * createQueue (void);
bool enqueue (QUEUE * queue, void * itemPtr);
void printQueue (QUEUE* stack);
int main (void)
{
QUEUE * queue1;
QUEUE * queue2;
QUEUE * queue3;
int * numPtr;
int** itemPtr;
queue1 = CreateQueue();
queue2 = CreateQueue();
queue3 = CreateQueue();
int i = 11;
numPtr = (int*) malloc (sizeof(i));
*numPtr = i;
enqueue(queue1, numPtr);
```

```
i = 22;
numPtr = (int*)malloc(sizeof(i));
*numPtr = i;
enqueue(queue1, numPtr);
i = 33;
numPtr = (int*) malloc(sizeof(i));
*numPtr = i;
enqueue(queue1, numPtr);
i = 44;
numPtr = (int*)malloc(sizeof(i));
*numPtr = i;
enqueue(queue2, numPtr);
i = 55;
numPtr = (int*)malloc(sizeof(i));
*numPtr = i;
enqueue(queue2, numPtr);
i = 66;
numPtr = (int*) malloc(sizeof(i));
*numPtr = i;
enqueue(queue2, numPtr);
i = 77;
numPtr = (int*) malloc(sizeof(i));
*numPtr = i;
enqueue(queue3, numPtr);
i = 88;
numPtr = (int*)malloc(sizeof(i));
*numPtr = i;
enqueue(queue3, numPtr);
i = 99;
numPtr = (int*) malloc(sizeof(i));
*numPtr = i;
```

```
enqueue(queue3, numPtr);
printf("Queue 1:\n");
printfQueue(queue1);
printf("Queue 2:\n");
printfQueue(queue2);
printf("Queue 3:\n");
printfQueue(queue3);
return 0;
}
QUEUE *createQueue(void)
{
QUEUE *queue;
queue = (QUEUE*)malloc(sizeof(QUEUE));
if(queue)
{
queue->front = NULL;
queue->rear = NULL;
queue->count = 0;
}
return queue;
}
bool enqueue(QUEUE* queue, void *itemPtr)
{
QUEUE_NODE *newPtr = (QUEUE_NODE*) malloc(sizeof
(QUEUE_NODE));
newPtr->dataPtr = itemPtr;
newPtr->next = NULL;
if(queue->count ==0)
queue->front = newPtr;
else
```

```
91  queue->rear->next = newPtr;
92  (queue->count)++;
93  queue->rear = newPtr;
94  return true;
95  }
96  QUEUE * destroyQueue (QUEUE *queue)
97  {
98  QUEUE_NODE* deletePtr;
99  if (queue)
100 {
101 while (queue->front != NULL)
102 {
103 free(queue->frontPtr);
104 deletePtr = queue->front;
105 queue->front = queue->front->next;
106 free(deletePtr);
107 }
108 free (queue);
109 }
110 return NULL;
111 }
112 void printQueue (QUEUE * queue)
113 {
114 QUEUE_NODE * node = queue->frount;
115 printf("Front =>");
116 while (node)
117 {
118 printf ("%3d", *(int*)node->datePtr);
119 node = node->next;
120 }
```

```
121    Printf("<= Rear\n");
122    return;
123  }
```