```
#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;
queue 1 = createQueue();
 queue 2 = createQueue();
queue3 = CreteQueue();
 Int j=1;
numPtr = (int*) malloc(sizeof(1));
* numptr =1;
enqueue (queue1, numPtr);
1=4;
return 0;
 }
```

```
numPtr = (int*) malloc (sizeof(i));
* numPtr = i;
enqueue(queue1, numPtr);
i = 6
numPtr = (int*) malloc (sizeof(i));
* numPtr = i;
enqueue (queue1, numPtr);
i = 2;
numPtr = (int*) malloc (sizeof(i);
* numPtr = i;
enqueue (queue2, numPtr);
i = 7;
numPtr = (int*) malloc (sizeof(i);
* numPtr = i;
enqueue (queue2, numPtr);
i = 3
numPtr = (int*) malloc (sizeof(i));
* numPtr = i;
enqueue (queue3, numPtr);
i = 8;
numPtr = (int*) malloc (sizeof(i)); * numPtr = i;
enqueue (queue3, numPtr);
i = 10;
numPtr = (int*) malloc (sizeof(i));
* numPtr = i;
enqueue (queue3, numPtr);
printf("queue 1:\n");
printQueue (queue1);
printf ("Queue 2:\n");
printf (" queue 3:\n");
prlnQueue (queue 3);
```

```
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*
{
QUEUE-NODE* NewPtr = (QUEUE_NODE*) malloc (sizeof
(QUEUE_NODE));
NewPtr -> dataPtr = itemPtr;
newPtr -> Next = NULL;
if (queue -> count == 0)
queue -> front = newPtr;
else.
queue -> rear -> Next = newPtr;
( queee -> count) ++;
queue -> rear = newPtr;
return ture;
}
QUEUE* destroyQueue(QEUEUE* queue)
{
QUEUE_NODE * deleteptr;
if (queue)
{
while (queue > front != NULL)
{
```

```
free(queue->front->dataPtr);
deleteptr = queue->front;
queue->front = queue->front->next;
free(deleteptr);
}
free(queue);
}
return NULL;
}
void printQueue(QUEUE *queue)
{
QUEUE_NODE *node = queue->front;
printf("Front=>");
while(node)
{
printf("%3d", *(int*)node->dataPtr);
node = node->next;
}
printf("<=Rear\n");
}
```