

CS 2103

Assignment in Data Structures # 6

(**basic Queue functions for Singly and Doubly Linked Queues**)

Submitted by:

**Robert Michael Lim, BSCS – 2**

Submitted to:

**Ms. Christine F. Peña**

/\*Singly Linked Queues\*/

typedef struct nodeType{

char elem;

struct nodeType \*next;

}\*nodePtr;

typedef struct{

nodePtr front, rear;

}Queue;

void initQueue(Queue \*Q)

{

Q->front = Q->rear = NULL;

}

void Enqueue(Queue \*Q, char x)

{

nodePtr temp;

temp = (nodePtr)malloc(sizeof(struct nodeType));

if(temp!=NULL){

temp->elem = x;

if(Q->front==NULL){

Q->front=Q->rear= temp;

}else{

Q->rear->next = temp;

Q->rear = temp;

}

}

Q->rear->next = NULL;

}

void Dequeue(Queue \*Q)

{

nodePtr temp;

if(Q->front!=NULL){

temp = Q->front;

Q->front = Q->front->next;

free(temp);

}

}

char Front(Queue Q)

{

return Q.front->elem;

}

/\*Doubly Linked Queues\*/

typedef struct node{

char elem;

struct node \*prev, \*next;

}\*Queue, nodeType;

void initQueue(Queue \*Q)

{

(\*Q) = NULL;

}

void Enqueue(Queue \*Q, char x)

{

Queue temp;

if((\*Q)==NULL){

(\*Q) = (Queue)malloc(sizeof(nodeType));

(\*Q)->prev = (\*Q)->next = (\*Q);

}else{

temp = (Queue)malloc(sizeof(nodeType));

if(temp!=NULL){

(\*Q)->prev->next = temp;

temp->prev = (\*Q)->prev;

(\*Q)->prev = temp;

temp->next = (\*Q);

}

}

(\*Q)->prev->elem = x;

}

void Dequeue(Queue \*Q)

{

Queue temp;

if((\*Q)->next == (\*Q)){

(\*Q) = NULL;

}else{

temp = (\*Q);

(\*Q)->next->prev = (\*Q)->prev;

(\*Q)->prev->next = (\*Q)->next;

(\*Q) = (\*Q)->next;

free(temp);

}

}

char Front(Queue Q)

{

return Q->elem;

}

In this particular assignment, I only had a few troubles with the Doubly Linked Queues. The trouble is that it is really tricky to access sometimes. If there is one thing I have learned in this assignment it is on how I can do so much when drawing especially on Doubly Linked Queues.