```
//
//  queue.c
//  All
//
//  Created by Filippo Fontanelli on 02/04/11.
//  Copyright 2011 __MyCompanyName__. All rights reserved.
//

#include "queue.h"

int init(queue *q, int *want_opt) {
    if (q != NULL) {
        q->want_opt = want_opt;
        q->cnt = EMPTY;
        q->length_string = EMPTY;
        q->tot_space = EMPTY;
        q->front = NULL;
        q->rear = NULL;
    } else
        return ERROR;

    return 0;
}

int init_token(token *t) {
    int i,j;
    if (t != NULL) {

        MALLOC(t->stringa, sizeof(char)*DEFAULT_NUM_COLUM)

        for (j = 0;j< 2;j++){
            for (i = 0; i< NUM_FORMATTAZIONI;i++)
                t->formattazione[i][j] = storico_formattazione[i][j];
        }
        MALLOC(t->link, sizeof(char)*LINK)
        t->n_space = 0;
        t->next = NULL;
    } else
        return ERROR;

    return 0;
}

int enqueue(token *t,queue *q) {
    int i,j;
    if (q == NULL) return ERROR;
    if (!empty(q)) {
        q->rear->next = t;
        q->rear = t;
    } else
        q->front = q->rear = t;
    q->cnt++;
    q->length_string+= strlen(t->stringa);
    q->tot_space+= t->n_space;
    for (j = 0;j< 2;j++){
        for (i = 0; i< NUM_FORMATTAZIONI;i++)
            storico_formattazione[i][j] = t->formattazione[i][j];
    }

    return 0;
}

token* dequeue(queue *q) {
    if (q == NULL) return NULL;
```



```
    token *p;
    p = q->front;
    q->front = q->front->next;
    q->cnt--;

    return p;

}

token* front(const queue *q) {
    if (q == NULL) return NULL;
    return q->front;
}

token* rear(const queue *q) {
    if (q == NULL) return NULL;
    return q->rear;

}

int empty(const queue *q) {
    return (q->cnt == EMPTY);
}

int full(const queue *q) {
    return (q->cnt == FULL);
}

int free_token(token* t) {
    if( t != NULL){
        free(t->stringa);
        free(t->link);
        free(t);
    }else
        return ERROR;
    return 0;
}

int free_queue(queue *q) {
    if( q != NULL){
        while(!empty(q)) {
            free_token(dequeue(q));
        }
        free(q);
    }else
        return ERROR;
    return 0;

}
```