#include <stdio.h>

#include <stdlib.h>

typedef int DataType;

typedef struct LinkQueueNode {

DataType data;

struct LinkQueueNode \*next;

} LinkQueueNode;

typedef struct {

LinkQueueNode \*front;

LinkQueueNode \*rear;

} LinkQueue;

void InitLinkQueue(LinkQueue \*LQ) {

LQ->front = LQ->rear = NULL;

}

int IsEmptyQueue(LinkQueue \*LQ) {

return LQ->front == NULL;

}

void EnterLinkQueue(LinkQueue \*LQ, DataType value) {

LinkQueueNode \*newNode = (LinkQueueNode \*)malloc(sizeof(LinkQueueNode));

if (newNode == NULL) {

printf("Memory allocation failed\n");

return;

}

newNode->data = value;

newNode->next = NULL;

if (LQ->rear == NULL) {

LQ->front = LQ->rear = newNode;

} else {

LQ->rear->next = newNode;

LQ->rear = newNode;

}

}

int DeleteLinkQueue(LinkQueue \*LQ, DataType \*value) {

if (IsEmptyQueue(LQ)) {

printf("Queue is empty\n");

return 0;

}

LinkQueueNode \*temp = LQ->front;

\*value = temp->data;

LQ->front = LQ->front->next;

free(temp);

if (LQ->front == NULL) {

LQ->rear = NULL;

}

return 1;

}

int GetHeadData(LinkQueue \*LQ, DataType \*value) {

if (IsEmptyQueue(LQ)) {

printf("Queue is empty\n");

return 0;

}

\*value = LQ->front->data;

return 1;

}

int GetLinkQueueLength(LinkQueue \*LQ) {

int length = 0;

LinkQueueNode \*current = LQ->front;

while (current != NULL) {

length++;

current = current->next;

}

return length;

}

void PrintLinkQueue(LinkQueue \*LQ) {

LinkQueueNode \*current = LQ->front;

while (current != NULL) {

printf("%d ", current->data);

current = current->next;

}

printf("\n");

}

void ClearLinkQueue(LinkQueue \*LQ) {

while (LQ->front != NULL) {

LinkQueueNode \*temp = LQ->front;

LQ->front = LQ->front->next;

free(temp);

}

LQ->rear = NULL;

}

void test\_Queue\_Link(void) {

DataType value;

LinkQueue \*LQ = (LinkQueue \*)malloc(sizeof(LinkQueue));

if (LQ == NULL) {

printf("Memory allocation failed\n");

return;

}

InitLinkQueue(LQ);

EnterLinkQueue(LQ, 1);

EnterLinkQueue(LQ, 2);

EnterLinkQueue(LQ, 3);

printf("Queue elements: ");

PrintLinkQueue(LQ);

value = GetLinkQueueLength(LQ);

printf("Queue length: %d\n", value);

GetHeadData(LQ, &value);

printf("Front element: %d\n", value);

DeleteLinkQueue(LQ, &value);

printf("Deleted element: %d\n", value);

printf("Queue elements: ");

PrintLinkQueue(LQ);

ClearLinkQueue(LQ);

free(LQ);

}

int main() {

test\_Queue\_Link();

return 0;

}

Now ，现在我写出来的版本

注意那串注释的代码

#include <stdio.h>

#include <stdlib.h>

typedef int DataType;

typedef struct linkqueuenode{

DataType data;

struct linkqueuenode \*next;

}linkqueuenode;

typedef struct{

linkqueuenode \*front;

linkqueuenode \*rear;

}linkqueue;

void initqueue(linkqueue \*lq){

lq->front=lq->rear=NULL;

}

int isemptyqueue(linkqueue \*lq){

return lq->front==NULL;

}

void enterlinkqueue(linkqueue \*lq,DataType value){

linkqueuenode \*newnode=(linkqueuenode\*)malloc(sizeof(linkqueuenode));

if(newnode==NULL){

printf("memory allocation failed\n");

return;

}

newnode->data=value;

newnode->next=NULL;

if(lq->rear==NULL){

lq->front=lq->rear=newnode;

}else{

lq->rear->next=newnode;

lq->rear=newnode;

}

}

int deletelinkqueue(linkqueue \*lq,DataType \*value){

if(isemptyqueue(lq)){

printf("queue is null\n");

return 0;

}

// linkqueuenode \*p=lq->front;

// \*value=p->data;

// lq->front->next=p->next;

// if(lq->rear==p){

// lq->rear=lq->front;

// }

// free(p);

linkqueuenode \* p=lq->front;

\*value=p->data;

lq->front=lq->front->next;

free(p);

if(lq->front==NULL){

lq->rear=NULL;

}

return 1;

}

int getheaddata(linkqueue \*lq,DataType \*value){

if(isemptyqueue(lq)){

printf("queue is null\n");

return 0;

}

\*value=lq->front->data;

return 1;

}

int getqueuelength(linkqueue \*lq){

int length=0;

linkqueuenode \*i=lq->front;

while(i!=NULL){

length++;

i=i->next;

}

return length;

}

void printlinkqueue(linkqueue \*lq){

linkqueuenode \*i=lq->front;

while(i!=NULL){

printf("%d",i->data);

i=i->next;

}

printf("\n");

}

void clearqueue(linkqueue \*lq){

while(lq->front!=NULL){

linkqueuenode\*temp=lq->front;

lq->front=lq->front->next;

free(temp);

}

lq->rear=NULL;

}

void test\_queue(){

DataType value;

linkqueue \*lq=(linkqueue\*)malloc(sizeof(linkqueue));

if(lq==NULL){

printf("memory allocation failed\n");

return;

}

initqueue(lq);

enterlinkqueue(lq,1);

enterlinkqueue(lq,3);

enterlinkqueue(lq,5);

printf("queue elements:");

printlinkqueue(lq);

value=getqueuelength(lq);

printf("queue length:%d",value);

getheaddata(lq,&value);

printf("first element:%d",value);

deletelinkqueue(lq,&value);

printf("delete element:%d",value);

printf(" after queue elements :");

printlinkqueue(lq);

clearqueue(lq);

free(lq);

}

int main(){

test\_queue();

return 0;

}