第二周周记

这是培训的第二周，整一个星期还是沉浸在代码的海洋当中无法自拔，先是别的工作室未完成的网页，又到周末的QG大礼包，这一周各种时间都已经被这两样工作所占据，真的挺烦的。总是在担心加入任务未完成会怎么样。这一周的大组培训，作业感觉通过了上一周的折磨之后熟悉了许多，知道了怎么建立一个工程文件，生成exe文件，也明白了到底是个什么要求，可能唯一的苦难就是面对新的数据结构怎么去理解的问题，栈这个数据还是比较陌生的，大概就知道有个先入后出的原理，别的其实也不怎么知道。到百度上搜索了，了一下别人是怎么写的操作，深入的理解了一下整个操作的过程，看着看着，貌似明白了其中的原理。然后尝试着不看自己去写出来。开始的时候也有卡住的地方，动态分配内存没有分配正确，导致程序总是运行不了，挺头疼的，后来再看看别人怎么写，让后自己再尝试着写写，发现成功了，心中一顿欢喜，后来的操作也是行云流水，作业这次的效率明显比上次大有提高，毕竟所有东西都熟悉了怎么操作。这个星期也没有干些什么别的事情，基本有时间就在完成工作室的任务，感觉生活瞬间失去了乐趣，开玩笑的哈哈。

这个星期学到的东西还是挺多的，首先是学到了一个新的数结构栈，然后接下来一系列与栈相关的操作。然后知道了怎么去使用git这个东西，看上去好像很高大上，输一串根本不知道什么意思的命令。然后知道师兄推荐的一系列对前端学习有用的网站。

第一点是基于链表的栈的一些操作：

Status pushLStack(LinkStack \*s,ElemType data)

{

LinkStackPtr p;

p=(LinkStackPtr )malloc(sizeof(StackNode));

p->data=data;

p->next=s->top->next;

s->top->next=p;

s->count=s->count+1;

return SUCCESS;

} //入栈

入栈的时候开始我理解为，他的头结点也必须给他赋值，发现原来不是，实在top的下一个next开始进行给数据域赋值，并不是直接在top赋值，top充当一个领头的作用，包括在进行删除操作的时候，也是这样的。

Status LStackLength(LinkStack \*s,int \*length)

{

if(s->top->next==NULL)

return ERROR;

else{

\*length=s->count;

}

return SUCCESS;

}

第二点印象比较深的，当参数传入一个整形变量的地址时，在主函数中应该怎么去调用它，要先同样传进一个整形指针的参数int \*k=&d;同时还是让他赋值为一个整形变量的取地址，最后通过，用整形变量从而实现把整形指针改变的值给调用出来。

基于数组的栈：

typedef struct SqStack {

ElemType \*elem;

int top;

int size;

} SqStack;

在定义的时候elem其实就是指数组，开始的时候不太明白这个elem其实做什么用的，后来才知道，它和链栈的区别在于，用户先要定住这个数组有多大的size，当用的时候不够再追加，而链栈大小变化比较灵活，不需要，预先就把它设定死，这也是他的好处。

Status stackLength(SqStack \*s,int \*length)

{

if(s->top!=-1){

\*length =(s->top+1);

return SUCCESS;

}

if(s->top==-1)

return ERROR;

} //检测栈长度

在使用这个数组栈的时候，操作和我们在平常数组当中有一点相似，都是通过下标法来引用数组当中的元素。

到现在其实我还没知道macdown是怎么使用的，先用着word写一写，之后会再去学习。