QG训练营第一周周记

从星期四接到了研发中心考核任务，到星期五QG训练营的培训。我就已经知道接下来的一个学期并不会过的那么得轻松了。当然了，你要想成为比一般人更加优秀得人必须得先付出比别人更加多得努力对吧，总而言之，即使再苦，也苦不过高三这两年得时光（我是一个复读生，本属于17届的）。

接到任务的周末，既带有一些烦恼也带有一些兴奋，兴奋是可能是因为刚进入到训练营的一种无知的冲动，烦恼是因为这些任务真的不轻松，这个周末因为有一些私人小事情回了一趟家。但是因为知道任务繁重，研发的网页制作，QG的数据结构，即使回到了家丝毫也不敢懈怠。星期六回到家直接干了一天，把网页制作完成了一半工程，数据结构的解题思路也经本形成。但是到了星期天晚上会到来才发现，自己做出来的东西完全不符合要求。以为在自己电脑编译器上能跑便是完成任务。过去都是基本所有东西塞在一个文件中运行，对工程文件，什么接口之类的完全不知道是什么，看了群里同学师兄的讨论也是一知半解的，大概只知道要自己写一个头文件拿到main函数中能运行就好，也不懂什么叫做接口，看了下QG发过来的文件也不知道有什么作用，因为在家不方便与同学讨论，便按照自己觉得的要求去做，结果GGGGGG了。

星期天晚上7点钟把自己原本的程序调试成功看了下同学，才发现原来要这么操作，我顿时惊呆了，一个周末做了无用功。当时已经是7点钟了，因为GCC生成工程文件操作我不会，其实也不知道到底可不可以，马上找同学拷贝了他的编译器，重新去做。但是拷贝过来之后，那个编译器我根本不会操作，一点也不熟悉，看着QG代码中一大堆的填空题，就知道自己已经离去世不远了。

星期天晚上还没做完作业强行熟悉了一下编译器，打了一下代码，一点用的没用，各种红字，错误一大堆。没办法了只能等第二天，能补多少补多少。到了星期一，白天课拉满，到了4点多回到宿舍重新开始写，这时候进度基本为0，打了半天，才发现传进去的是2级指针，之前链表都是1级指针操作，根本不会2级指针，各种百度，各种问同学，看到了晚上7点还没有成功运行过一次的，心态瞬间爆炸，搞到了9点钟实在不行了，越搞头越痛，休息了下，到了11点才看，可能真的当时头太晕了，浪费了不少时间，心态又炸裂，想着作业可能交不了，到了11点休息过后，再上机，发现思路有一点打开了，运行也有成功，通过前面几个小时对这个2级指针的熟悉，慢慢能操作起来，搞到半夜还是要休息。

星期二通过了星期一的摸索过程，程序慢慢有了眉目，利用一些课后时间终于实现了第一个单链表。

问题：1.在第一次QG训练营作业当中，明显发现C语言真的很弱很弱，二级指针怎么操作当时一看不熟悉，函数指针的使用，也不太熟悉，怎么样多函数多文件编程也不会。

2.对于一些很细节的地方比如什么时候该停止循环，循环条件把握不准确，整天导致程序运行崩溃，然而自己又不知道错在哪里。

3.对于枚举体内容不熟悉。刚开始都不知道原来enum时枚举体类型。以及当他做函数类型的时候返回值应该怎么储存下来。

4.对于typeof的声明含义不清晰。

5.对一些经典算法如插入逆转，只停留在理论上，只知道思路，并没有真正操作过，导致实践时出了无限问题。

这次作业下来真的是发现漏洞百出，真的是流下没技术的泪水。直到今天终于完成单链表之前心情真的无比烦躁，今天课都没听得进去，就一直在想代码怎么敲，

收获：既然问题多收获当然也很大

1. ```typedef struct LNode {

ElemType data;

struct LNode \*next;

} LNode, \*LinkedList;

```

对于这种声明时间上是用LNode代替了struct得命名，Linkedlist 代替了指向struct得一级指针，所以定义Linkedlist \*p就是一个二级指针。

2. ```\*L=(LinkedList)malloc(sizeof(LNode)); ```

但传入一个二级指针，我们要给他动态分配内存的时候也先在其前面进行解引用，同时注意两个（）里的书写，第一个是类型名，\*L之后变成一级指针固要写Linkedlist，绝对不能写Linkedlist \*，第二个括号是结构体的size，千万别写成Linkedlist

3. ```·void TraverseList(LinkedList L, void (\*visit)(ElemType e)) ```

形参是函数指针，函数名称为visit，返回类型为空类型，使用的时候直接visit(p->data);即可，但是在使用时要先定义后这种类型函数功能才可以使用。

4. ```Status ReverseList(LinkedList \*L) {

int i;

LinkedList \*current=L,\*pre,\*temp;

(\*pre)=NULL;

while((\*current)!=NULL)

{

\*temp=(\*current)->next;

(\*current)->next=\*pre;

\*pre=\*current;

\*current=\*temp;

}

\*L=\*pre;

return SUCCESS;

}

```

利用二级指针对链表进行逆转，必须先对二级指针进行\*操作，才能操作。

5.

```

LinkedList p,mid;

mid = p= \*L;

while (p->next != NULL)

{

if (p->next->next!=NULL)

{

p = p->next->next;

mid = mid->next;

}

else

p = p->next;

}

return mid;

}

```

快慢指针法寻找链表中间节点，核心一个指针步长为1，另一个指针步长为2；最后返回中间节点地址。

6. ```

Status InitList(LinkedList \*L) {

\*L=(LinkedList)malloc(sizeof(LNode));

(\*L)->next=NULL;

return SUCCESS;

}

```

返回类型为枚举体，主函数中要用一个int型数据将其接住，枚举内容默认顺序从0开始逐渐增大。

7.最重要的东西就是知道了工程文件时怎么操作的，head.h文件中放你所需要的函数声明，另一个。C文件中写实现函数的功能，在main函数中添加你的头文件，调用你的函数。