SM2 21st Basic C Programming

Second Checkpoint Fake Test

Solutions and Papers

Lovingly created by 全群最菜

Question 1:

你听说过利滚利吗… 假如我们每个月在银行结算利息之后把钱取出来再存回去，（理论上）存款的增长速度会越来越快，给你初始存款a，月利率b，求需要多少个月你的存款会达到目标c.

以函数int savings(double start\_money, double rate, double target\_money)返回需要的月份数。

检查答案的办法：在计算器上先输入start\_money\*（rate+1），然后按Ans\*(rate+1)，数你按过多少次等号之后结果不小于target\_money。

**int** savings(**double** a, **double** rate, **double** tar){  
 **static int** mon = 0;  
 **if**(a>=tar) **return** mon;  
 **else**{  
 mon++;  
 **return** savings(a\*(1+rate),rate,tar);  
 }  
}

(0 marks)

Question 2:

聪明的ygy是一个很有程序设计天赋的人，与此同时他的英语也强的一比，于是他想要把自己的十万英语词汇压缩成一个字符串；他心想，Prof Tay曾经说过一个叫做sub-string的东西，于是他希望自己的所有英语词汇都是这个新字符串的sub-string，这样对于mislead和lead和advantages和gesture就可以合成misleadvantagesture这么一个新字符串了！

YGY觉得自己要给其他人留一点活路，所以他并不要求你找到一个最短的这样的字符串，你只需要从头到尾把他给你的这些单词拼起来就行了！

Sub-quest 2.1:

创作一个函数int overlap(char a[], char b[])，输出单词a的末尾几个字母是单词b的开头；比如mislead和leading重复的部分就是lead，该函数就应该返回4。当然这个答案可能不唯一，比如walalalalala和lalalala这两个单词，重复部分可以是la, lala, lalala, lalalala，你应该给出最长的那个重复长度——在这个例子里面就是8——毕竟聪明的ygy想要尽可能压缩字符串长度。

如果两个单词根本就不能拼在一起，说明ygy当时有点迷糊，请先打印“YGY is the strongest!”再返回-1。

**int** overlap(**char** a[], **char** b[]){  
 **int** lena = strlen(a), lenb = strlen(b);  
 **int** ans = 0, i, mini, count;  
 **char** \*p1,\*p2;  
 mini = lenb < lena ? lenb : lena;  
 **for**(i=1;i<=mini;i++){  
 p1 = &a[lena-i]; p2 = b;  
 count = 0;  
 **while**(count <= i && \*p1 != **'\0'** && \*p2 != **'\0'** && \*p1 == \*p2){  
 count++; p1++; p2++;  
 }  
 **if**(count == i) ans = i;  
 }  
 **if**(ans == 0){  
 printf(**"YGY is the strongest!"**);  
 **return** -1;  
 } **else return** ans;  
}

(1 mark)

Sub-quest 2.2:

创作一个函数int stitch(char buff[], char a[], char b[])，在确定单词a和单词b可以拼起来的前提下，把拼接后的字符串放入buff[]字符串中；拼接成功返回1，否则返回0。

**int** stitch(**char** buff[], **char** a[], **char** b[]){  
 **int** len = overlap(a,b);  
 **int** i,j;  
 **if**(len == -1) **return** 0;  
 **else**{  
 **for**(i=0;i<strlen(a);i++)  
 buff[i] = a[i];  
 **for**(j=0;j<=strlen(b)-len;j++){  
 buff[i] = b[len+j];  
 i++;  
 }  
 **return** 1;  
 }  
}

(1 mark)

Finale:

以void smart\_ygy(char buff[], char input[][11], int count)的形式给出你的答案，其中count是ygy给你的单词个数，约定每个单词长度不大于10（包括terminator不超过11个字符），打印出最终的字符串。

//都到这个份了，还写不出来？

(98 marks)

END OF PAPER, PLEASE 想干嘛干嘛