

# 《计算机程序设计》作业 No-12 及第 11 次上机

作业内容要点： 结构体和链表

【 姓名 \_\_\_\_\_ 学号 \_\_\_\_\_ 】

(一)在计算机上编程序，加上**必要的注释**。

(二)上机实验，经助教检查通过后，复制**源码**并记录**实验结果**，完成**报告**。

(三)实验报告：记录**调试及改错过程**；知识点或方法技巧的**收获心得**。



1. 利用恺撒加密方法，设偏移量为 7，将下面的文本文件 `secret.txt` 加密存储，覆盖掉原文内容。

【注】在密码学中，恺撒密码（英语：Caesar cipher），或称恺撒加密、恺撒变换、变换加密，是一种最简单且最广为人知的加密技术。它是一种替换加密的技术，明文中的所有字母都在字母表上向后（或向前）按照一个固定数目进行偏移后被替换成密文。例如，当**偏移量**是 3 的时候，所有的字母 A 将被替换成 D，B 变成 E，以此类推。这个加密方法是以罗马共和时期恺撒的名字命名的，当年恺撒曾用此方法与其将军们进行联系。

具体方法可参考 <https://baike.baidu.com/item/%E6%81%BA%E6%92%92%E5%AF%86%E7%A0%81/4905284>

`secret.txt` 文件内容：

In physics, a quantum (plural quanta) is the minimum amount of any physical entity (physical property) involved in an interaction. The fundamental notion that a physical property can be "quantized" is referred to as "the hypothesis of quantization". This means that the magnitude of the physical property can take on only discrete values consisting of integer multiples of one quantum.

(一) 【源码】

```
1. //  
2. // main.c  
3. // 121201
```

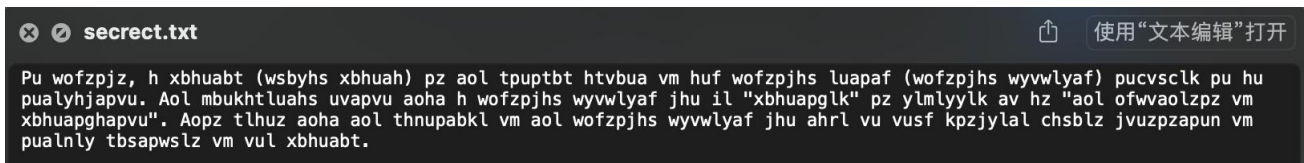
```
4. //
5. // Created by 李佩哲 on 2021/12/15.
6. //
7.
8. #include <stdio.h>
9. #include <stdlib.h>
10. #include <string.h>
11.
12. typedef struct str{
13.     char s;
14.     struct str *next;
15. } STR;
16.
17. STR *secret(STR *head){
18.     for (STR*p=head; p!=NULL; p=p->next) {
19.         if((p->s<='s'&&p->s>='a')||(p->s<='S'&&p->s>='A'))p->s+=7;
20.         else if (p->s>'s'&&p->s<='z')p->s+='a'+6-'z';
21.         else if (p->s>'S'&&p->s<='Z')p->s+='A'+6-'Z';
22.         else continue;
23.     }
24.     return head;
25. }
26.
27. int main(int argc, const char * argv[]) {
28.     STR *head, *node, *end;
29.     head=(STR*)malloc(sizeof(STR));
30.     end=head;
31.     FILE *fp=fopen("/Users/page/Documents/-文稿/1-学习/1-理学/5-计算机科学/2-C/Homework/All_Homework_c/121201/secret.txt", "r", stdin);
32.     for (STR*p=(STR*)malloc(sizeof(STR)); (scanf("%c",&p->s)!=EOF); ) {
33.         node=(STR*)malloc(sizeof(STR));
34.         node->s=p->s;
35.         end->next=node;
36.         end=node;
```

```

37. }
38. end->next=NULL;
39. head=secret(head);
40. fclose(fp);
41. FILE *fp1=fopen("/Users/page/Documents/-文稿/1-学习/1-理学/5-计算机科学/2-C/Homework/All_Homework_c/121201/secret.txt", "w", stdin);
42. for (STR*p=head; p!=NULL; p=p->next) {
43.     fprintf(fp1,"%c",p->s);
44. }
45. fclose(fp1);
46. return 0;
47.}

```

## (二) 【运行结果】



## (三) 【实验报告】

freopen 用 r 打开文件，用链表读取字符，每个字母各自+7，然后再 freopen 一个 w 文件去覆盖原来的

2. 有一个“乐谱”文件“music.txt”如下，请编程序演奏这个乐谱。

【说明】文件每一行给出的是两个数据，分别表示一个声音的频率（Hz）和播放的时间长度（ms）。播放声音可以使用 Beep()函数。这个 WindowsAPI 函数原型为：

```

BOOL Beep(
    DWORD dwFreq,

```

## DWORD dwDuration

);

这里 DWORD 类型即 unsigned int, 参数 dwFreq 是频率值, 取值范围是 37 到 32,767 (0x25 – 0x7FFF). 参数 dwDuration 是播放时长, 单位毫秒。例如, *Beep(392,375);*

详见: <https://docs.microsoft.com/en-us/windows/win32/api/utilapiset/nf-utilapiset-beep>

### “乐谱”文件 MUSIC.TXT 内容:

392 375

392 125

440 500

392 500

523 500

494 1000

392 375

392 125

440 500

392 500

587 500

523 1000

392 375

392 125

784 500

659 500

523 500

494 500

440 1000

689 375

689 125

659 500

523 500

587 500

523 1000

### (一) 【源码】

```
1.//
2. // main.c
3. // 121202
4. //
5. // Created by 李佩哲 on 2021/12/14.
6. //
7.
8. #include <stdio.h>
9. #include <stdlib.h>
10. #include <windows.h>
11.
12. typedef struct music{
13.     int Hz;
14.     int time;
15.     struct music *next;
16. } MUSIC;
17.
18. void play(MUSIC*head){
19.     MUSIC *p;
20.     for (p=head; p!=NULL; p=p->next) {
21.         Beep(p->Hz,p->time);
22.     }
23.}
24.
25. int main(int argc, const char * argv[]) {
26.     FILE *fp=fopen("D:/121202/MUSIC.txt", "r", stdin);
27.     MUSIC *head, *node, *end;
```

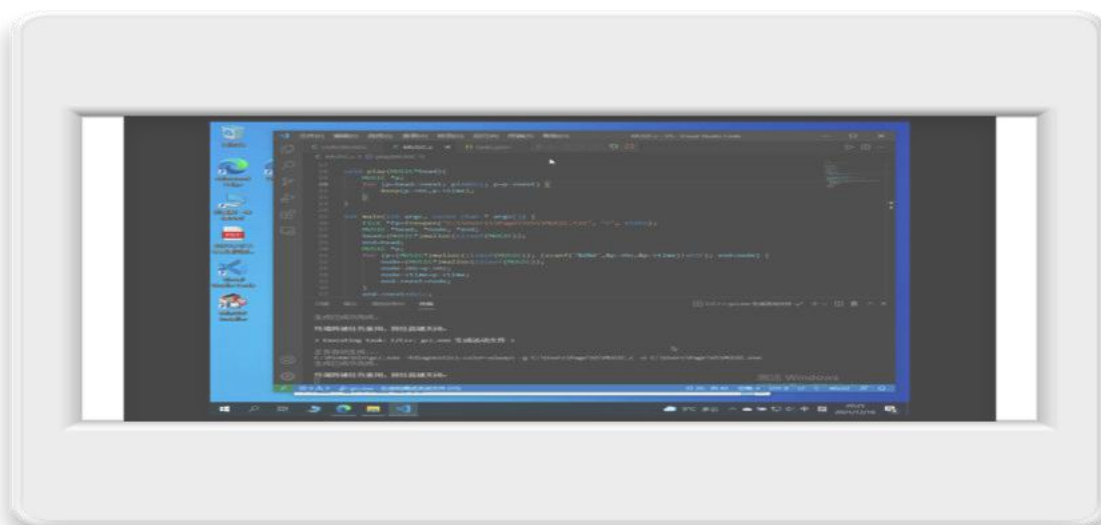
```

28. head=(MUSIC*)malloc(sizeof(MUSIC));
29. end=head;
30. MUSIC *p;
31. for (p=(MUSIC*)malloc(sizeof(MUSIC)); (scanf("%d%d",&p->Hz,&p->time)!=EOF); end=node) {
32.     node=(MUSIC*)malloc(sizeof(MUSIC));
33.     node->Hz=p->Hz;
34.     node->time=p->time;
35.     end->next=node;
36. }
37. end->next=NULL;
38. play(head);
39. fclose(fp);
40. return 0;
41. }

```

## (二) 【运行结果】

视频如下（双击打开）：



## (三) 【实验报告】

用链表存储频率和时长，每个节点存储一个音符

[注]另外, 参考下面的声音频率表, 你也可以自己“作曲”。

### Frequency

C	261.6
C#	277.2
D	293.7
D#	311.1
E	329.6
F	349.2
F#	370.0
G	392.0
G#	415.3
A	440.0
A#	466.2
B	493.9
C	523.2