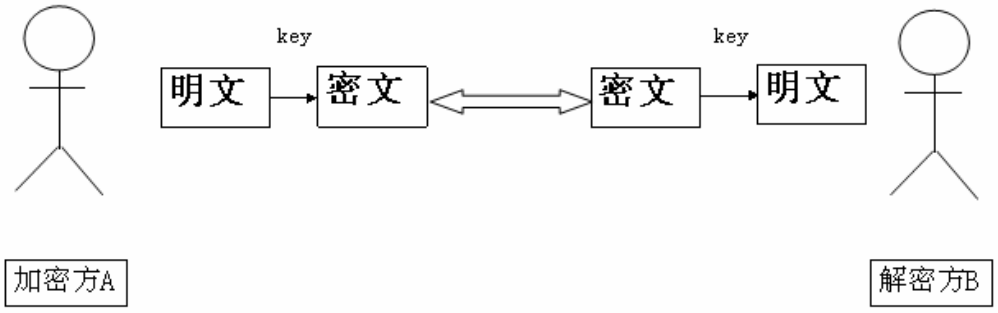
文件加密/解密

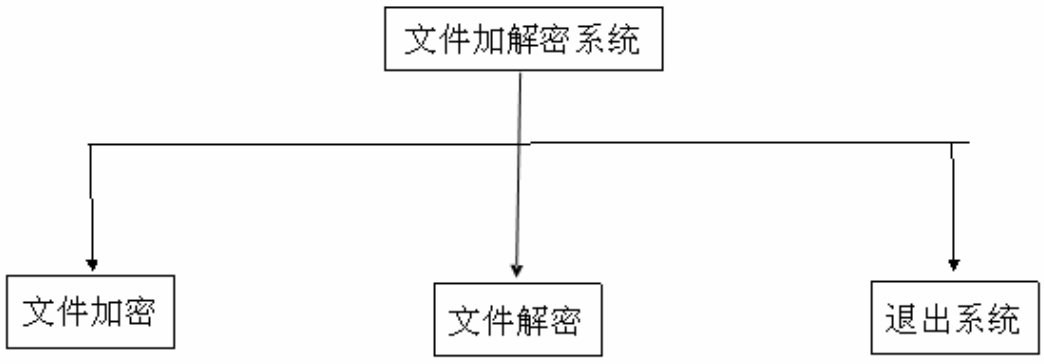
题目分析:

对称加密体制是传统而经典的加密体制策略。所谓对称加密体制即加密方 A 和解密方 B 共享一个密钥 key。加密方 A 使用该密钥 key 对要保密的文件进行加密操作，从而生成密文；解密方 B 同样使用该密钥 key 对加密方生成的加密文件实施解密操作，从而生成明文。



对称密码体制示意

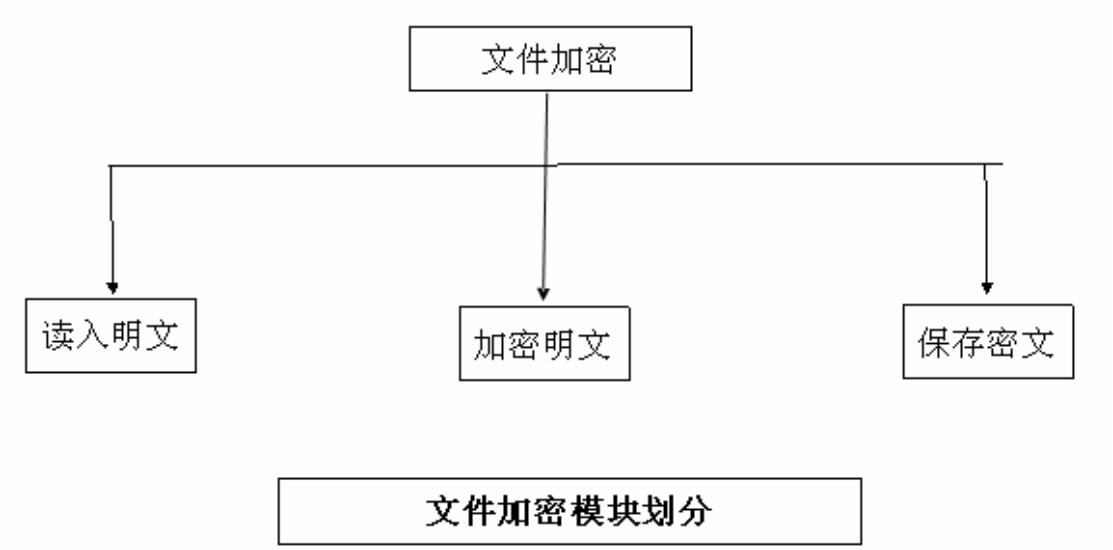
二：设计概要



文件加密/解密系统的结构功能图

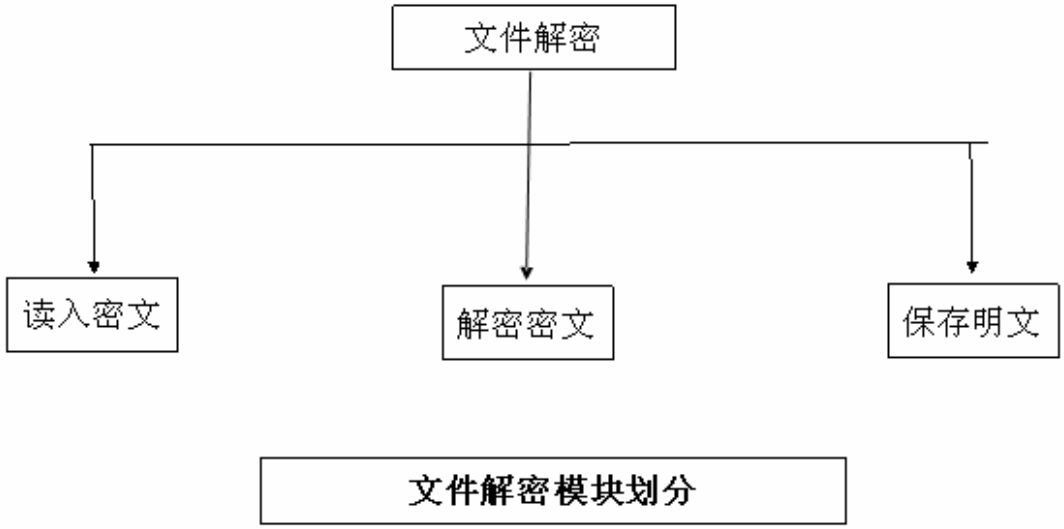
文件加密和文件解密模块又可以继续向下划分：

文件加密：



文件加密模块划分

文件解密：



文件解密模块划分

项目提示：

1：打开文件的时候用二进制方式打开进行读写。

2：测文件大小方法：

1）用 fseek() 定位流指针到文件的末尾。

2）用 ftell()函数测流指针的位置即文件的大小。

3：读文件内容：

1）根据文件的大小用 malloc 申请内存空间保存读出的内容

2）读文件数据的时候要从文件的开始读（rewind（））。

详细设计 API 设计参考：

1) 从键盘获取源文件和目的文件名字

```
/*****************************************************************************/
//函数功能:获取 目的文件和源文件的名字
//参数： src_file_name:源文件名字字符数组首地址。
// dest_file_name:目的文件的名字字符数组首地址
/*****************************************************************************/
void get_file_name(char * dest_file_name,char * src_file_name)
```

2) 从文件中读出内容

```
/*****************************************************************************/
//函数功能:读出文件内容
//参数：file_length:整型指针，此地址中保存文件字节数。
// src_file_name:文件名字，从此文件中读取内容。
// 返回值:读出字符串的首地址
// 在此函数中测文件的大小，并 malloc 空间，再把文件内容读出返回，读出字符数组的首地址
/*****************************************************************************/
char * read_src_file(unsigned long int *file_length,char *src_file_name)
```

3) 字符数组加密

```
/*****************************************************************************/
//函数功能:加密字符串
//参数：
//        src_file_text:要加密的字符串。 length:字符串的长度
//        password: 加密密码
// 返回值: 加密后的字符串的首地址
// 加密原理字符数组中每个元素加上 password
/*****************************************************************************/
char * file_text_encrypt(char * src_file_text,unsigned long int length,unsigned int password)
```

4) 解密字符串

```
/*****************************************************************************/
//函数功能:解密字符串
//参数：
//        src_file_text:要解密的字符串。 length:字符串的长度
//        password: 解密密码
// 返回值: 解密后的字符串的首地址
//思想;把数组中的每个元素减去 password 给自己赋值。
/*****************************************************************************/
char * file_text_decrypt(char * src_file_text,unsigned long int length,unsigned int password)
```

5) 保存文件

```
/*****************************************************************************/
//函数功能:将字符串保存到目的文件中
//参数：
//        text:要保存的字符串首地址
//        file_name :目的文件的名字
//        length:字符串的长度
//思想：传入字符数组的首地址和数组的大小及保存后的文件的名字，即可保存数组到文件中
/*****************************************************************************/
void save_file(char* text,unsigned long int length,char * file_name)
```

6) 打印文件信息

```
/*****************************************************************************/
//
//        函数功能:打印帮助信息
//
//
/*****************************************************************************/
void print_help()
{
        printf("********1:加密文件**********\n");
        printf("********2:解密文件**********\n");
        printf("********3:退出程序**********\n");
}
```