一、 实践内容

（一）实践的主要内容：

①分别实现 ECB、CBC、CFB、OFB 这四种操作模式的 DES。每种操作模式都有一组对应的测试数据，以便检查程序的正确性。其中，CFB 操作模式为 8 位 CFB 操作模式，OFB 操作模式为 8 位 OFB 操作模式。

②以命令行的形式，指定明文文件、密钥文件、初始化向量文件的位置和名称、加密的操作模式以及加密完成后密文文件的位置和名称。

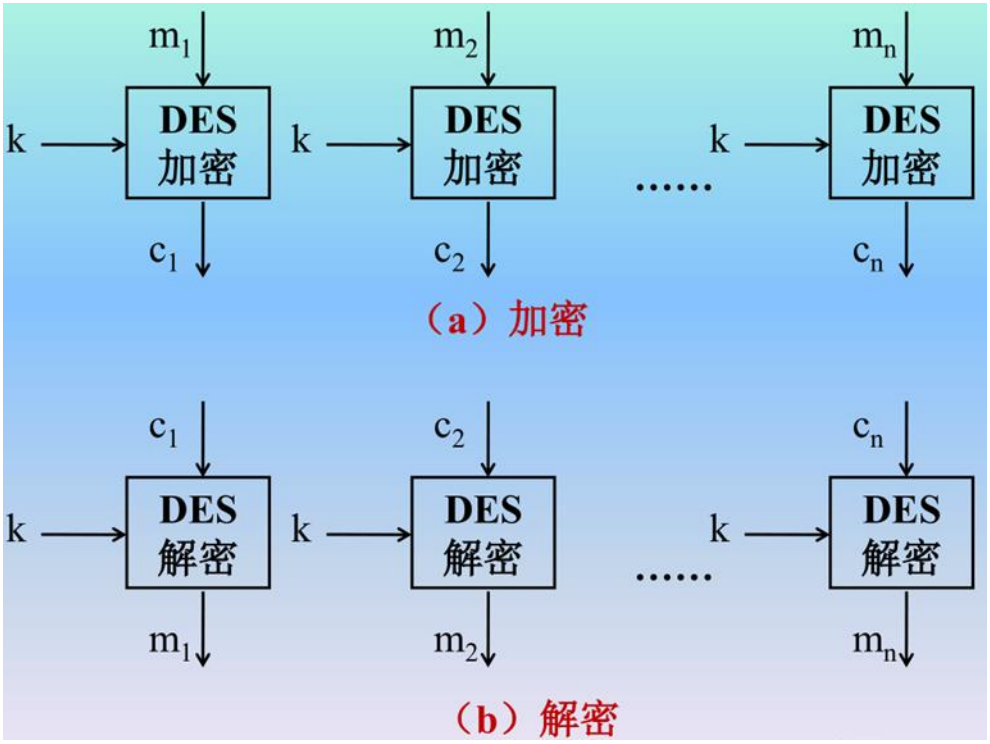
③分别实现对每种操作模式下加密及解密速度的测试，要求在程序中生成 5MB 的随机测试数据，连续加密、解密 20 次，记录并报告每种模式的加密和解密的总时间和速度。

（二）相关原理：

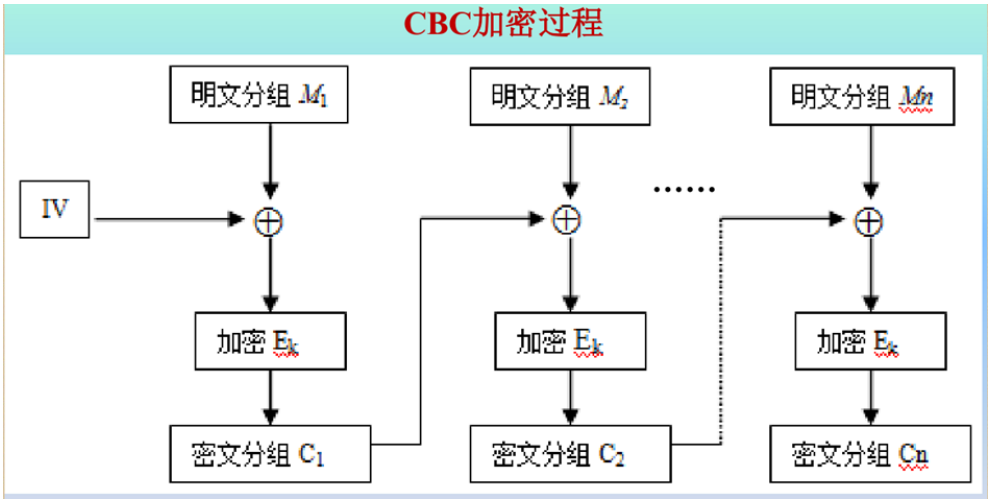
①DES 是一种明文分组 64bit，有效密钥 56bit，输出密文 64bit，具有 16 轮迭代的分组密码算法。在本次课程实践中，有一部分是要实现对一个数据块（即一个明文分组--64bit）的 DES 加密。DES 由初始置换、16 轮迭代、初始逆置换组成。

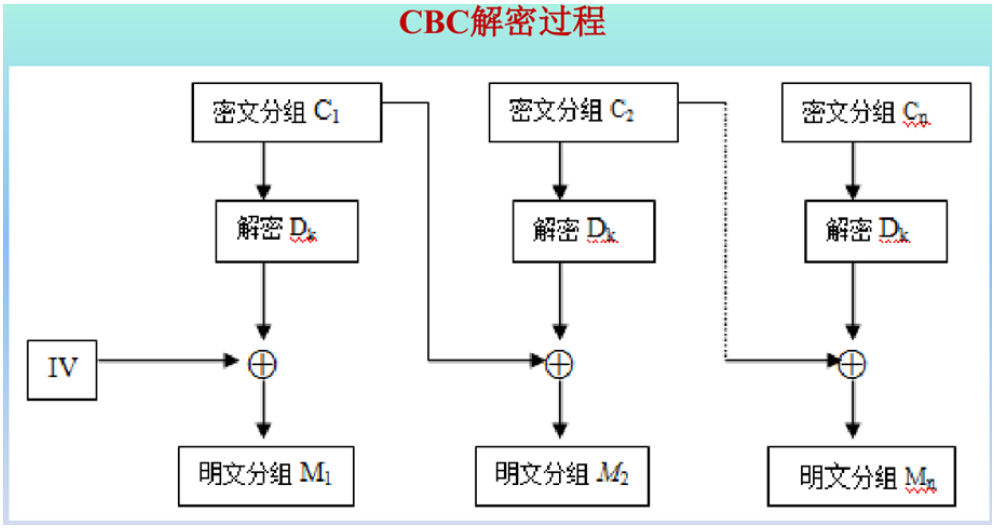
②分组加密应用于大数据加密，就牵涉到了各种不同的模式。在本次课程实践中，要求实现的是 ECB\CBC\CFB\OFB 四种模式的应用。

ECB：将每块明文加密成相应的密码块，若最后一块不足 64bit，则用一些任意二进制序列填充。相同明文块会被加密成相同密文块。

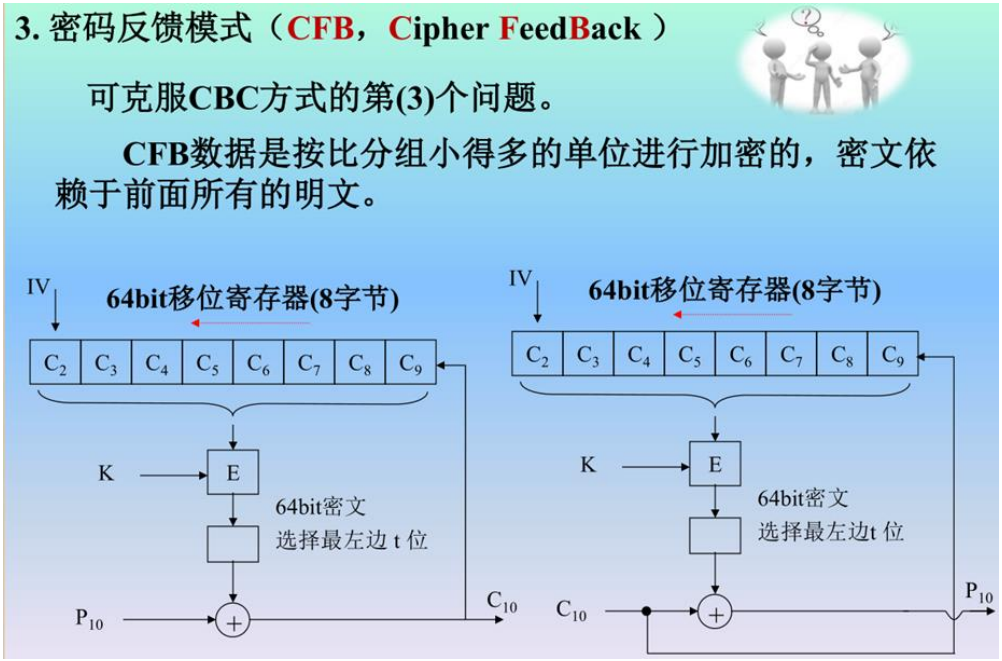


CBC：加入反馈机制，当前明文块先与前面的密文块进行异或，再加密。

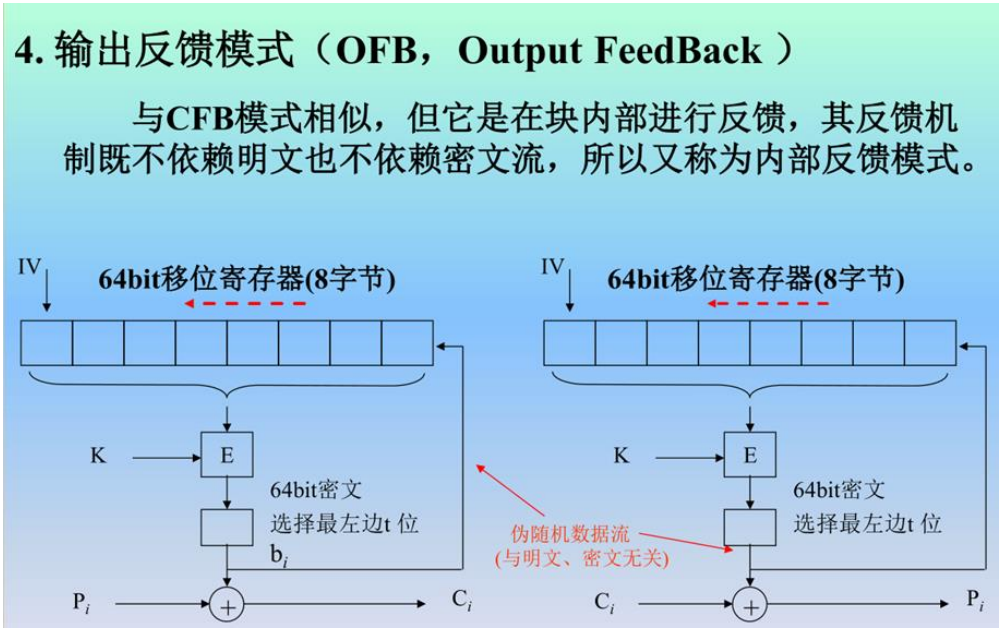




CFB：按比分组小得多的单位进行加密（使用 64bit 移位寄存器）。将前一个分组的密文加密，再和当前分组的明文进行异或。



OFB：与 CFB 类似，但是是在块内部进行反馈，通过将明文分组和密码算法的输出进行异或来产生密文分组，反馈机制不依赖明文和密文流。



③DES 属于 Feistel 结构，它的 F 函数不要求可逆，加、解密算法结构相同，只需要每一轮的子密钥反过来使用。

二、 实践环境

pc 操作系统：win10

代码编写及编译 IDE：codeblocks

执行:命令行运行编译产生的 exe 文件

编程语言：C 语言

三、 实践过程与步骤

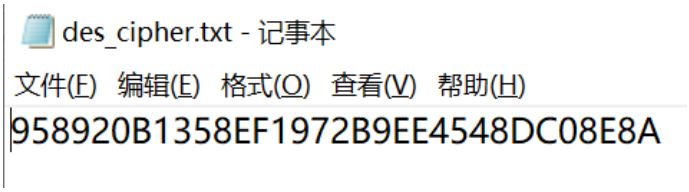
（一） 实现 ECB、CBC、CFB、OFB 这四种操作模式的 DES，用一组对应的测试数据检查程序的正确性：

①ECB：支持输入 9 个参数或 11 个参数（仍旧给出 iv 文件，但实际不会用到）。

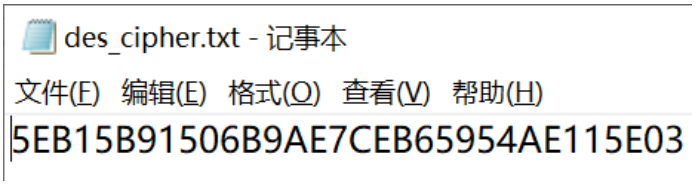
输入 9 个参数：

输入 11 个参数：

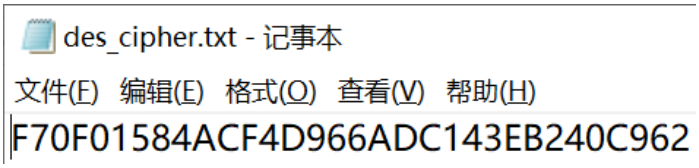
密文也会输出到指定 txt 文件，如下所示：



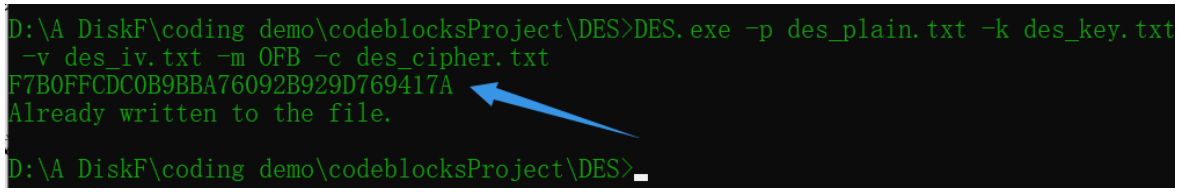
②CBC：输入 11 个参数。

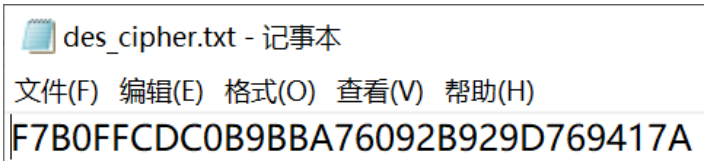


③CFB：输入 11 个参数。



④OFB：输入 11 个参数。





（二） 在程序中生成 5MB 的随机测试数据，连续加密、解密 20 次，记录并报告每种模式的加密和解密的总时间和速度：

5MB 的数据即为 5*(10^6)*8bit，由于在程序实现时，一个十六进制字符对应 4bit，因此在程序中需要随机生成的数据（十六进制字符 0x00 到 0x0F）是 10M 个。

在这一部分实践中，代码改动自上一部分即可，增加了加解密模式的自动转换和计时功能。通过程序生成 10M 个字符数据写入明文文件，然后进行加解密，循环 20 次，查看运行时间结果。

①ECB：20 次循环总用时 1032955.000000ms，5MB 数据一次加解密用时 51647.75ms，速度为（5*2）/51.64775=0.1936MByte/s：

```
D:\A DiskF\coding demo\codeblocksProject\DES_speed>DES_testSpeed.exe -p des_plain.tx
t -k des_key.txt -m ECB -c des_cipher.txt
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.

20 rounds ALL DONE.
The time spent is 1032955.000000ms.
```

②CBC：20 次循环总用时 1301209.000000ms，5MB 数据一次加解密用时 65060.45ms，速度为（5*2）/65.06045=**0.1537MByte/s**：

```
D:\A DiskF\coding demo\codeblocksProject\DES_speed>DES_testSpeed.exe -p des_plain.tx
t -k des_key.txt -v des_iv.txt -m CBC -c des_cipher.txt
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.

20 rounds ALL DONE.
The time spent is 1301209.000000ms.
```

③CFB：20 次循环总用时 8802534.000000ms，5MB 数据一次加解密用时 440126.7ms，速度为（5*2）/440.1267=**0.0227MByte/s**：

```
D:\A DiskF\coding demo\codeblocksProject\DES_speed>DES_testSpeed.exe -p des_plain.tx
t -k des_key.txt -v des_iv.txt -m CFB -c des_cipher.txt
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.

20 rounds ALL DONE.
The time spent is 8802534.000000ms.
```

④**OFB**：20 次循环总用时 8625385.000000ms，5MB 数据一次加解密用时 431269.25ms，速度为（5*2）/431.26925=**0.0232MByte/s**：

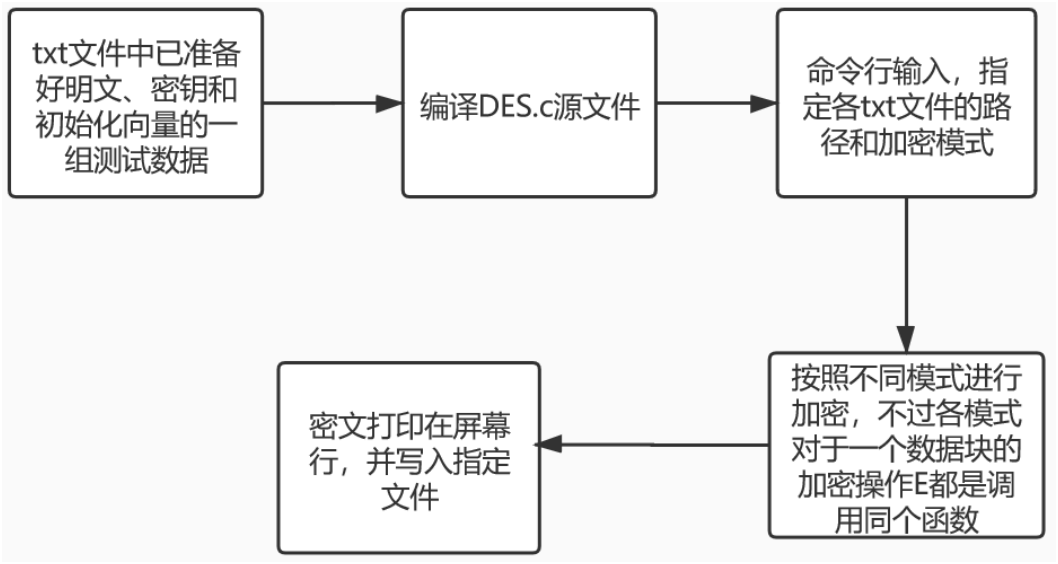
```
D:\A DiskF\coding demo\codeblocksProject\DES_speed>DES_testSpeed.exe -p des_plain.tx
t -k des_key.txt -v des_iv.txt -m OFB -c des_cipher.txt
No.1 Encryption done.   No.1 Decryption done.
No.2 Encryption done.   No.2 Decryption done.
No.3 Encryption done.   No.3 Decryption done.
No.4 Encryption done.   No.4 Decryption done.
No.5 Encryption done.   No.5 Decryption done.
No.6 Encryption done.   No.6 Decryption done.
No.7 Encryption done.   No.7 Decryption done.
No.8 Encryption done.   No.8 Decryption done.
No.9 Encryption done.   No.9 Decryption done.
No.10 Encryption done.   No.10 Decryption done.
No.11 Encryption done.   No.11 Decryption done.
No.12 Encryption done.   No.12 Decryption done.
No.13 Encryption done.   No.13 Decryption done.
No.14 Encryption done.   No.14 Decryption done.
No.15 Encryption done.   No.15 Decryption done.
No.16 Encryption done.   No.16 Decryption done.
No.17 Encryption done.   No.17 Decryption done.
No.18 Encryption done.   No.18 Decryption done.
No.19 Encryption done.   No.19 Decryption done.
No.20 Encryption done.   No.20 Decryption done.

20 rounds ALL DONE.
The time spent is 8625385.000000ms.
```

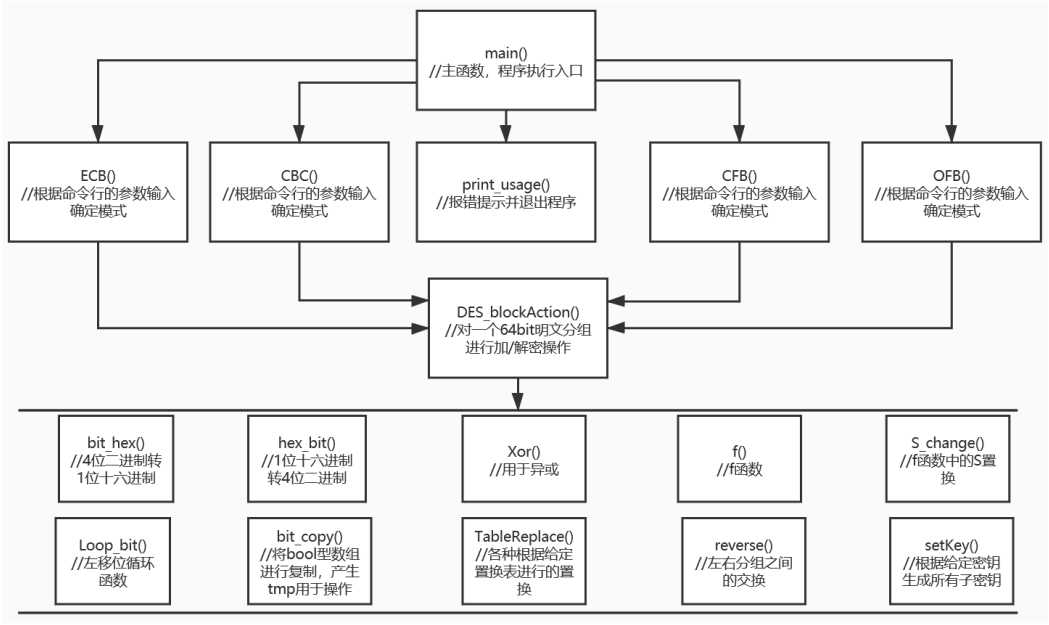
四、 程序设计方案

本程序涉及到了文件读写，在同一目录下提前已准备好

des_plain.txt/des_key.txt/des_iv.txt 和 des_cipher.txt 四个文件。前三个文件中都是有一组测试数据的，加密结果打印到屏幕上，同时也会写入 des_cipher.txt。具体的程序流程为：



此外，在 C 语言设计的程序中，通常都会涉及到众多函数的调用。下面我通过自己定义的函数的调用层次示意图来展现整个程序（不包含库函数，只画出自己写的功能性函数）：



具体代码见 C 源文件 *DES.c*。

五、 实践结果与分析

实验结果截图参见上文第四部分——实践过程与步骤。

对于实践的第一部分，经过验证，对于指定的测试数据，各个模式的加密结果都是正确的，这表明程序编写正确且成功执行了。

对于测试速度的第二部分实践内容，根据结果可见 ECB/CBC 模式的加解密速度相接近，CFB/OFB 的加解密速度相接近；ECB/CBC 的速度大约是 CFB/OFB 的 7~8 倍。这是合理的，因为在本次实践中，CFB 操作模式为 8 位 CFB 操作模式，OFB 操作模式也是 8 位 OFB 操作模式，这意味着明文实际上也是 8 位一组了，分组变小，组数远远增多，加密操作及相关的异或操作次数都变为了 ECB/CBC 模式的 8 倍左右。因此，引入反馈的 ECB/CBC 模式的速度会是 CFB/OFB 模式的 8 倍左右，这与实际的实践结果一致。