Report1：

本次写的是给的第一个项目，也就是对reduced-SM3的birthday-attack，在本次实验中，我所使用的SM3实现方式是Python语言和基于GmSSL国密库的实现，也就是我在第一周实验课所研读的内容，在暑假期间加以完善和整理。在实现过程中，因为在python中输出的接口是bytes类型，因此我选择将它转化为16进制数，然后逐渐加大比较的位数，来测试找到碰撞的效果，对于碰撞的两个数，每次随机生成即可。找到的最好结果是找到了48位的碰撞，以下展示了我所找到的40位碰撞和48位碰撞：其中random1和random2分别是找到的两个碰撞的bytes形式，每次随机生成，msg1和msg2分别是它们的hash形式，以16进制显示，观察msg1和msg2的前几位也可以看到确实找到了40bit和48bit的碰撞。

```
C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\sm3.py"
完成啦!
random1 是:  b'\xbe\x90M\xb6\xbc\xde\xbe\x10\xc6\xa5\xc0\x87Q\xd9\xd03\xc89{\xa6J\xdb\x80\x00"\xa1\r\xaf\xf6\xe8O\x15'
random2 是:  b'_\x85%\x06Xcv\xe8\xe6\xd6=\xe5\xa0\xe5uA,:\x94\xdd\x9f\x14Xp\xdaT3\xfa\xd4\xb6\xa8\t'
msg1 是:  cf3f707dbc9b7e955b11be85f5bf7b727753068a6b5f936d04c66115843eb582
msg2 是:  cf3f7afc68940726146f4e5121a2f64bb3bc46268c5da00faf80213221e3b5cb
共花费时间:  510.62466406822205
```

以上是找到的40位碰撞

```
C:\Users\86180>set PYTHONIOENCODING=utf8 & C:\Users\86180\AppData\Local\Programs\Python\Python39\python.exe -u "c:\Users\86180\Desktop\sm3.py"
完成啦!
random1 是:  b'\xa6c\xce%\x064\x1e\xc3\x95\xc0\x94\xd1\xd7\xce\xb0\x1d{~\x8f\xd4Y\xa1\x8fxj\xcd,a\xaa\x9f\xac\xb0'
random2 是:  b'\x05a\xa9>3\xa6;e\xebQ\xa9\xb9\xb1\x18\xe6\xd2\x85\x94\x15\x8fn\xfd\x94@\xcd\xd6.Nz\x04Z\x88'
msg1 是:  4648e8b32da2588a0214f1f9094bf6d08367b958fb4ebf917e0df3e70e9ee062
msg2 是:  4648e8a84b2abe38e78c052d082a912c4d1be49336ffe5ad623a5425a5b1517c
共花费时间:  8502.489362478256
```

以上是找到的48位碰撞