

综合大作业报告

学号：

姓名：

专业：

写在最前：（程序清单）

z-01.doc 综合大作业报告；

z-01.pdf 综合大作业报告 pdf 版本

z-02.asm 完整输出公钥私钥，并对输入字符串进行加密解密的程序；

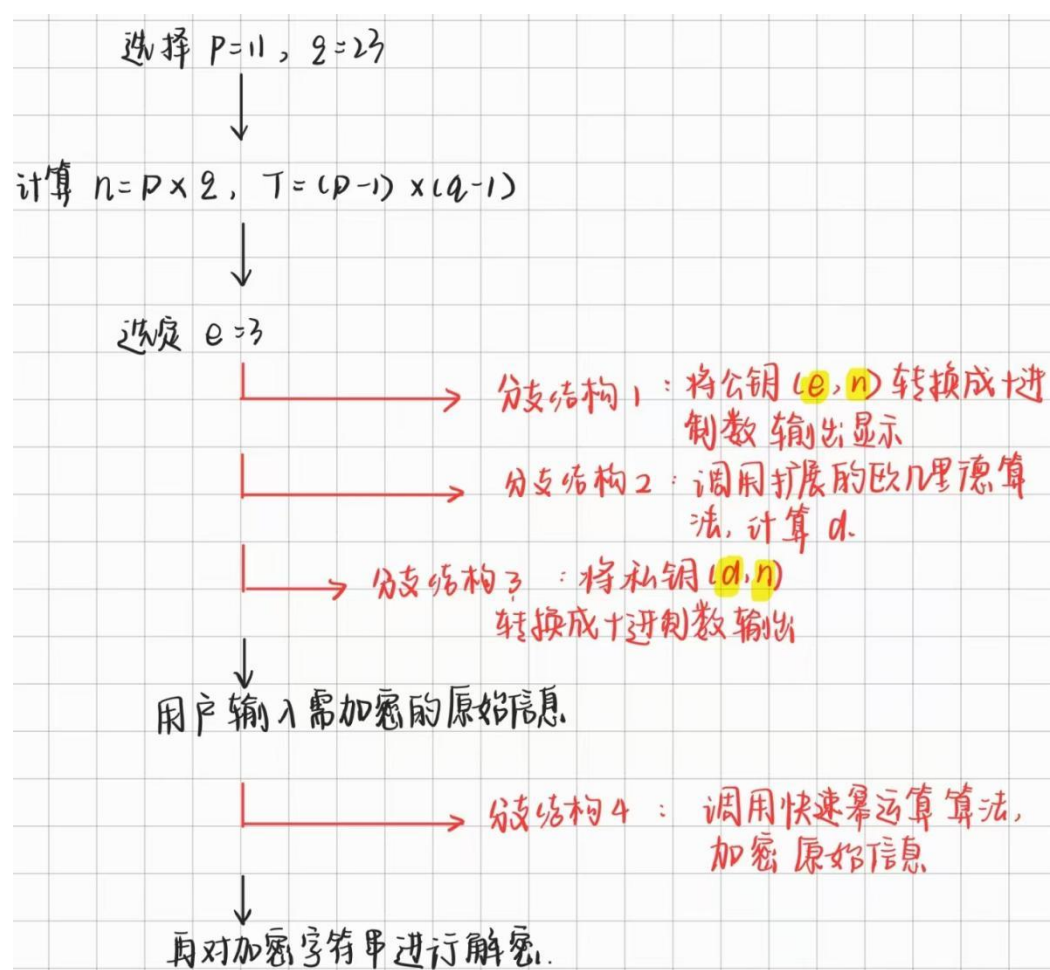
z-03.cpp 扩展的欧几里得算法 cpp 版，用于和汇编语言版对照；

在此次作业中，我采用分支结构来编写程序（但是编写程序的思想还是将整个任务分成几个子任务，分别编写子任务之后再统一在一起）

不采用主子程序的结构是因为在连接过程中，不同文件之间需要传输的参数太多，汇编总是出现问题，再就是我花费了大量时间查找资料解决这一问题但却没有成功，所以我采用了更简单的分支结构来编写程序，还希望老师可以见谅。

一、程序设计相关说明

- 程序流程图：



● 结合流程图程序说明如下:

首先选定 p 、 q 、 e ，计算 n 和 t

调用子程序 1--将 e 、 n 的值以十进制数的形式显示出来（因为 **DSOBox 0.74-3** 只能以 **ASCII 码** 的形式显示）

调用子程序 2--调用扩展的欧几里得算：计算 e 和 t 的最大公约数：
如果是互质，返回主程序继续运行；如果不是互质，则直接终止程序运行；计算贝祖等式，从而计算出私钥 d

调用子程序 1--将 d 、 n 的值以十进制数的形式显示出来

等待用户**输入**需要加密的字符串

调用子程序 3--调用快速幂算法，先求幂，再取余，得到加密后的

字符串

输出显示加密后的字符串（因为在 **DSOBox 0.74-3** 环境下，所以加密后的字符串是以 **ascii** 码对应的字符显示出来的，具体会在使用相关说明示例中解释）

输出显示解密后的字符串

二、程序使用相关说明

- 程序运行的硬件环境：基于 x86 架构的处理器
- 程序运行的软件环境：运用 MASM 汇编器及其对应的 linker 和 debugger
- 只需在 **DSOBox 0.74-3** 软件里运行即可
- 程序使用方法：

```
A:\>811.exe

the public key is:
e= 3
n= 253
the secret key is:
d=147
n= 253
The input string:
,,,
THE Encrypted STRING:
uuui
the decrypted string is:
,,,
```

811.exe 是测试程序，对应提交程序中的 z-02.asm

“the input string”之前的，均在程序一运行即出现，等待用户输入字符串

之后显示处理后加密解密的结果

- 输入的信息范围：

由于 dosbox 环境下，是读取用户输入字符的 **ascii** 码进行运算，所以很容易溢出，对照以下 **ascii** 码图表，可以看到，输入的信息范围

必须在 **28H** 以内。否则就会溢出，得到错误的加密信息。

表 2.7 标准 ASCII 码表

| $b_7b_6b_5b_4$ | 0 | 1 | 2 | 3 | 4 |
|----------------|-----|-----|----|---|---|
| $b_3b_2b_1b_0$ | | | | | |
| 0 | NUL | DLE | SP | 0 | @ |
| 1 | SOH | DC1 | ! | 1 | A |
| 2 | STX | DC2 | " | 2 | B |
| 3 | ETX | DC3 | # | 3 | C |
| 4 | EOT | DC4 | \$ | 4 | D |
| 5 | ENQ | NAK | % | 5 | E |
| 6 | ACK | SYN | & | 6 | F |
| 7 | BEL | ETB | ' | 7 | G |
| 8 | BS | CAN | (| 8 | H |
| 9 | HT | EM |) | 9 | I |
| A | LF | SUB | * | : | J |
| B | VT | ESC | + | ; | K |
| C | FF | FS | , | < | L |
| D | CR | GS | - | = | M |
| E | SO | RS | . | > | N |
| F | SI | US | / | ? | O |

计算器

程序员

28 × 28 × 28 =

FA00

HEX FA00

DEC 64,000

OCT 175 000

BIN 1111 1010 0000 0000

QWORD MS Mv

按位 位移位

A << >> CE ☒

B () % ÷

C 7 8 9 ×

D 4 5 6 −

E 1 2 3 +

● 出错信息的含义及注意事项:

```
A:\>811.exe

the public key is:
e= 3
n= 253
the secret key is:
d=147
n= 253
The input string:
,,,
THE Encrypted STRING:
uuui
the decrypted string is:
,,,
```

‘对应的加密信息是 u，至于末尾出现的符号是由于输入信息后按下回车键所导致。

但是只有按下回车键程序才可继续运行，目前本人所学无法避免这一问题。

程序设计中规定了缓冲区的大小，因此输入的字符不能超过 **10个**（包括最后的回车键）！

三、 调试相关说明

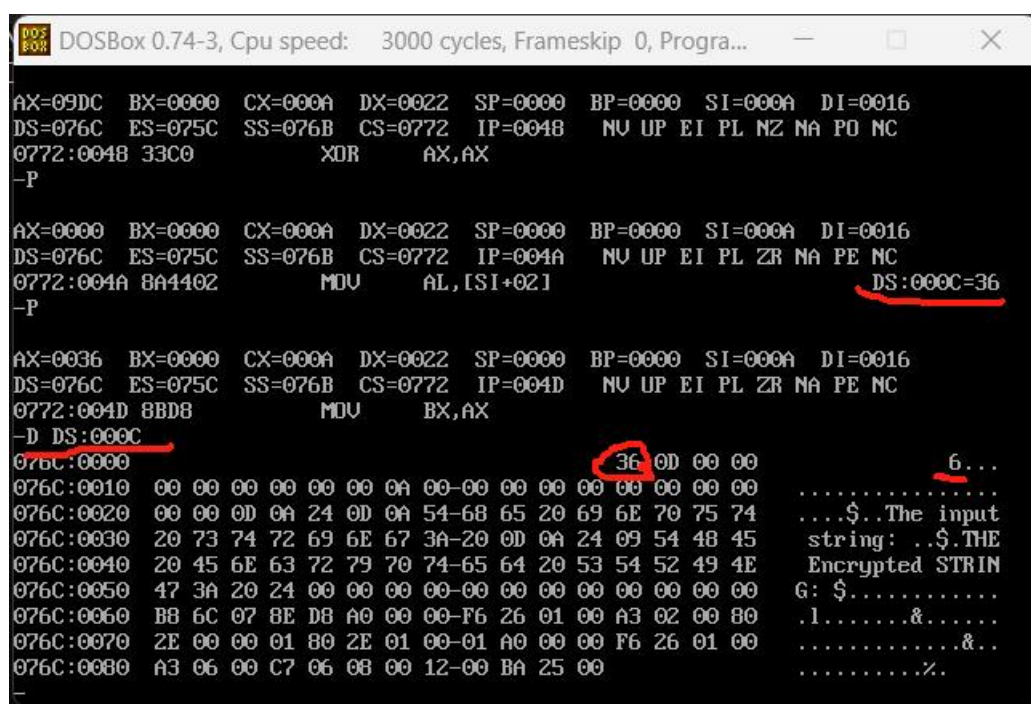
● 调试情况:

1. 上机遇到的问题及解决方法:

Q1:为什么会得到错误的加密的结果?

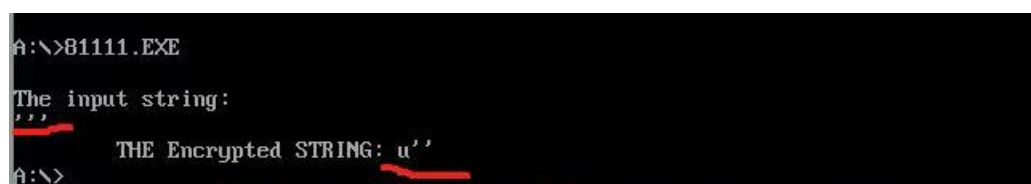
A1:1.第一次输入的 e 太大，加密过程中会导致数据溢出;

2.修改 e 的大小之后，输入某些信息会正确加密，某些信息会被错误加密--是因为从键盘里输入的字符会被处理为 **ascii 码** (见下图)，比如输入 6 后，存储在寄存器中就是 36H，再经过乘法运算后就很容易溢出 ($36H * 36H * 36H = 26718H$, 溢出)，因此也有了信息的处理范围。



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
AX=09DC BX=0000 CX=000A DX=0022 SP=0000 BP=0000 SI=000A DI=0016
DS=076C ES=075C SS=076B CS=0772 IP=004B NU UP EI PL NZ NA PO NC
0772:004B 33C0 XOR AX,AX
-P
AX=0000 BX=0000 CX=000A DX=0022 SP=0000 BP=0000 SI=000A DI=0016
DS=076C ES=075C SS=076B CS=0772 IP=004A NU UP EI PL ZR NA PE NC
0772:004A 8A4402 MOV AL,ESI+021 DS:000C=36
-P
AX=0036 BX=0000 CX=000A DX=0022 SP=0000 BP=0000 SI=000A DI=0016
DS=076C ES=075C SS=076B CS=0772 IP=004D NU UP EI PL ZR NA PE NC
0772:004D 8BDB MOV BX,AX
-D DS:000C
076C:0000 36 0D 00 00 6...
076C:0010 00 00 00 00 00 00 0A 00-00 00 00 00 00 00 00 .....
076C:0020 00 00 0D 0A 24 0D 0A 54-68 65 20 69 6E 70 75 74 ....$.The input
076C:0030 20 73 74 72 69 6E 67 3A-20 0D 0A 24 09 54 48 45 string: ..$.THE
076C:0040 20 45 6E 63 72 79 70 74-65 64 20 53 54 52 49 4E Encrypted STRIN
076C:0050 47 3A 20 24 00 00 00 00-00 00 00 00 00 00 00 G: $.
076C:0060 B8 6C 07 8E DB A0 00 00-F6 26 01 00 A3 02 00 80 .l.....&.....
076C:0070 2E 00 00 01 80 2E 01 00-01 A0 00 00 F6 26 01 00 .....&..
076C:0080 A3 06 00 C7 06 08 00 12-00 BA 25 00 .....%.
```

Q2:在测试程序中，输入的字符为什么没有完全被加密?



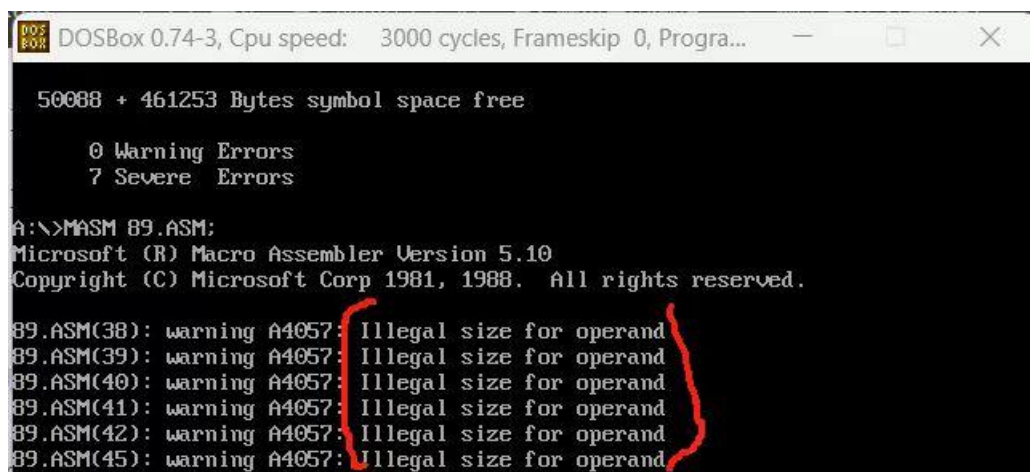
```
A:\>B1111.EXE
The input string:
'...'
THE Encrypted STRING: u''
A:\>
```

A2:调试检查之后发现循环次数不对，导致未能逐个读取输入的字符，加上一行，即可正确循环。

```
MOV     [DI+2],AL
INC     SI
INC     DI
LOOP    NEXT
```

```
INC     SI
INC     DI
MOV     E,3
LOOP    NEXT
```

Q3:汇编时出现下列问题怎么处理？



```
DOSBox 0.74-3, Cpu speed: 3000 cycles, Frameskip 0, Progra...
50088 + 461253 Bytes symbol space free
0 Warning Errors
7 Severe Errors
A:\>MASM 89.ASM:
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.
89.ASM(38): warning A4057: Illegal size for operand
89.ASM(39): warning A4057: Illegal size for operand
89.ASM(40): warning A4057: Illegal size for operand
89.ASM(41): warning A4057: Illegal size for operand
89.ASM(42): warning A4057: Illegal size for operand
89.ASM(45): warning A4057: Illegal size for operand
```

```
MOV     AA,BB
MOV     BB,RR
MOV     K1,K2
MOV     L1,L2
MOV     K2,KK
```

错误程序段：

A3:因为 mov 指令不能直接在两个内存中的自定义变量之间传递值，必须通过寄存器作为中间媒介。

```
mov     bx,bb
mov     aa,bx
mov     bx,rr
mov     bb,bx
mov     bx,k2
mov     k1,bx
mov     bx,l2
mov     l1,bx
mov     bx,kk
mov     k2,bx
```

修改后程序段：，即可成功运行。

Q4:C 语言程序运行后 得到的是 1，-73，私钥 d 是 1？还是-73？均不是，而是 147。为什么？


```
Microsoft Visual Studio 调试控制台
1
3 220
-73 1
C:\Users\HP\Desktop\汇编大作业\extend\x64\Debug\extend.exe (进程 21096) 已退出，代码为 0。
按任意键关闭此窗口。 . . .
```

(c 程序显示解释：第一行“1”表示接下来只输入一组数据即可；第二行则为输入的一组数据；第三行是处理得到的结果。)

A4:原因是扩展的欧几里得算法是通过解贝祖等式得到的系数，但满足贝祖等式的系数并不唯一，上述 c 程序只是显示一组系数结果；而 RSA 算法中对密钥 **d** 的规定是取逆反元素（详情可见提交的参考文献）。

如何通过扩展的欧几里得算法得到正确的密钥呢？

---辗转相除时必须用 $e \div t$ ($t=(p-1)*(q-1)$) ($e < t$) **[经过多次试验得到的结论]**

Q5:为什么处理后的信息有?

```
A:\>811.EXE
0
The input string:
'
    THE Encrypted STRING: u!
A:\>811.EXE
0
The input string:
''''
    THE Encrypted STRING: uuuu!
```

A5:是由于输入信息后按下回车键（CR）所导致。但是只有按下回车键程序才可继续运行，目前本人所学无法避免这一问题。

Q6:为什么会出现以下问题？


```
A:\>MASM 811.ASM:
Microsoft (R) Macro Assembler Version 5.10
Copyright (C) Microsoft Corp 1981, 1988. All rights reserved.

811.ASM(25): error A2028: Operator expected
49970 + 457274 Bytes symbol space free

0 Warning Errors
1 Severe Errors
```

A6:

```
BUFFER1 DB 10,0,10 DUP(0)
BUFFER2 DB 10,0,10 DUP(0)
LINE DB 13,10,"$"
msg1 DB 13,10,'The input string: ',13,10,'$'
msg2 db 13,10,'THE Encrypted STRING: ',13,10,'$'
msg3 db 'the secret key is:',13,10,'$'

DATA ENDS
```

少了个逗号。。。。。。

```
msg1 DB 13,10,'The input string: ',13,10,'$'
msg2 db 13,10,'THE Encrypted STRING: ',13,10,'$'
msg3 db 'the secret key is:',13,10,'$'
```

● 运行结果分析:

```
A:\>811.exe

the public key is:
e= 3
n= 253
the secret key is:
d=147
n= 253
The input string:
,,,
THE Encrypted STRING:
uuu!
the decrypted string is:
,,,
```

‘对应的加密信息是 u，至于末尾出现的符号是由于输入信息后按下回车键所导致。

但是只有按下回车键程序才可继续运行，目前本人所学无法避免这一问题。

● 样本数据设计:

对照以下 `ascii` 码图表，可以看到，输入的信息范围必须在 **28H** 以内。否则就会溢出，得到错误的加密信息。

The image shows a presentation slide with the title '表 2.7 标准 ASCII 码表'. The table lists ASCII values from 0 to 127, grouped by their hexadecimal representation (0-9, A-F). A red box highlights the first 16 characters (0-15), which correspond to the range 0-28H. To the right of the table is a screenshot of a Windows calculator in 'Programmer' mode, showing the hexadecimal value 'FA00'.

| b ₆ b ₅ b ₄ | 0 | 1 | 2 | 3 | 4 |
|----------------------------------------------|-----|-----|----|---|---|
| 0 | NUL | DLE | SP | 0 | @ |
| 1 | SOH | DC1 | ! | 1 | A |
| 2 | STX | DC2 | " | 2 | B |
| 3 | ETX | DC3 | # | 3 | C |
| 4 | EOT | DC4 | \$ | 4 | D |
| 5 | ENQ | NAK | % | 5 | E |
| 6 | ACK | SYN | & | 6 | F |
| 7 | BEL | ETB | ' | 7 | G |
| 8 | BS | CAN | (| 8 | H |
| 9 | HT | EM |) | 9 | I |
| A | LF | SUB | * | : | J |
| B | VT | ESC | + | ; | K |
| C | FF | FS | , | < | L |
| D | CR | GS | - | = | M |
| E | SO | RS | . | > | N |
| F | SI | US | / | ? | O |

● 判断样本数据的正确性:

计算逻辑：键盘键入字符——读取字符的 `ASCII` 码，以 16 进制存储在寄存器中——经过 $x^{e\%n}$ 的计算（均为十六进制的运算）——得到加密后的信息——以 `ascii` 码对应的字符显示出来。

示例：' (27h) —— $AX=0027$ —— ($e=3=0003h$; $n=253=00FDh$) —— $E7B7\%FD=75H$ —— 75H 是 'u' 的 `ASCII` 码，故结果显示为 u

四、课程总结

在暑假的两周，通过小学期，我认识了汇编语言。

作为一个纯小白，我在没有提前接触过 `c` 语言等任何语言的情况下（但是大学计算机的课程上我有接触到一点点 `SQL` 语句和 `python`，所以也算对计算机语言有一个非常浅显的了解），刚开始学习汇编语言，确实遇到了一些问题，而且别人两个小时可以完成的上机实验，我可能需要课下花一整个晚上才能真正消化整个程序的逻辑以及撰

写这类程序时需要注意的地方。

但这两周收获，远远大于正常教学周两周的收获。比如课上出现的一些 c 语言描述的东西，我课上不懂，但是我的同桌会给我讲，我课下之后也在慢慢学习 c 语言相关的课程；自己写的程序汇编时总是报错，因为知道我没有程序设计的基础，老师会帮我一行行看过去修改，教我自己去对比修改后和修改前的程序逻辑，建立基本的程序设计思路……所以这两周，除了对汇编的基本了解，能够编写简单的汇编程序并连接、运行之外，我学到了一些关于 c 语言以及程序设计的基本思路，这些收获对我之后的学习都大有帮助。谢谢老师和我的同桌。

在这两周的学习期间，上完课我就会去图书馆重新整理消化当天的学习内容，不懂的地方就及时问问身边的朋友；学习资料除了老师发的教材之外，川大的同学也给我提供了他们的课程 ppt 以及教材电子版，两者结合着学习，就基本能做到当天学习的内容当天掌握。同时 b 站（哔哩哔哩）上也有一些很好的课程可供日常学习参考。

但即使这样，由于课程时间的限制，现在我也只能写一些最基本最简单的程序，然后进行调试。如果是复杂一点的程序，就可能需要很长时间才能调试成功，就比如测试时写的程序，由于时间紧张，我未能正确调试好程序，也算是我考试时的一点遗憾吧，但这也说明了我能力的不足。但经过这次大作业的“洗礼”，我个人觉得我的调试能力到达了一个新的高度，毕竟这个程序完全是我自己写成，调试过程中出现的问题都是自己去研究解决的，目前基本可以做到调试程序

然后发现程序漏洞，从而修改。但是也还是有一些不足，比如由于没有系统学过 c，对 vs 的运用还不是很熟练，还不是很清楚如何利用 vs 去连接、运行汇编程序，所以这次的大作业我也只是采用最简单的分支结构去编写。

如果对我目前的编程能力进行自我评估的话，首先，我觉得我的编程能力肯定是属于比较初级的水平。目前可以对现有的程序框架进行补充完善和修改，但还是难以不借助任何参考资料地独立写出一个完整程序。

关于实验报告，虽然我不知道正确、标准的实验报告如何书写，但我拿出了端正的态度去独立完成这份作业，也尽可能地去详细阐述我的想法，尽可能地完善实验报告内容。如果有重复的地方，还请老师见谅。

最后是关于课程教学的一些改进意见。老师上课曾说，这个小学期不是让我们变成一个汇编专家，而是认识汇编，在之后用到汇编语言的时候可以快速上手。虽然如此，但是这个课程安排也还是逆天，说简单，又有点难；说难，也算不上多难。我上完的感觉是，课堂上好像只需要我了解就行，但是作业却需要我“精通”，而且很多的东西都需要自己课下去学习，然后再找老师去看。这样其实效率蛮低的。说了这么多个人感受，主要想说这个课程的内容和进度可以重新安排一下，确立一个明确的课程目标。（1279 字）

五、大作业总结

本次大作业设计，其实花了很长的时间。从一开始每天断断续续地找资料、看视频，了解 RSA 算法的基本原理，除此之外，由于我个人对数学比较感兴趣，也去了解了一下扩展的欧几里得算法的基本原理，便于我更好地去书写程序。由于搜索资料的时间非常零碎，基本上是有空了就搜一些，所以没有记录下很多参考资料的名目（结尾列出的是记录下的一些参考数目和网络资料链接）。不算搜索资料的时间，从我真正开始入手编写程序到大作业报告的完成，大概是五天左右（当然这五天里我还在完成军训任务），上机时数大概是 30 小时左右，上机地点在艺嘉楼（军训期间我在政治部办公室），均为独立完成。所以有一些完成不到位的地方还希望老师见谅。

关于课程的心得体会，我基本都写在课程总结那里了，这里就不再赘述。

关于大作业中存在的问题，我的程序能完成简单的信息加密，但是解密的过程，由于密钥 d 太大，特别容易溢出，所以解密过程我直接输出了缓冲区的内容，并不算真正意义上的信息解密；报告内容，我已经尽可能地在详细阐述，对照着老师的要求，一项一项地写过去了（虽然没有特别标明哪一段对应哪一个要求，但是确实是覆盖了所有的要求）；程序本身存在的不足的话，其实就是我前文反复提到过的，太容易溢出了，这个其实可以通过使用 32 位寄存器解决，但是我对于利用 vs（visual studio）进行 32 位程序的汇编还不是很能熟练掌握，我就依旧使用的是 16 位进行编写。

在理论和实践方面的总结的话，其实在接触汇编语言之前，就有

朋友跟我讲学码类专业肯定是离不开上手实操的，所以学习任何一门语言，理论和实践都是密不可分的。每次编写程序也好，调试也好，如果最后自己能通过运用理论所学，解决实际中遇到的问题，真的是一件非常有成就感的事情。

所以未来继续学习其他语言的话，我也依然不会忘记实践的重要性，积极上手实操。相信通过这次课程学到的程序设计的方法以及自己感悟到的一些学习计算机语言的心得技巧，会在我未来的学习中给予我莫大的帮助。

学习资料

[1]赵杰峰.基于 RSA 算法的网络信息加密方法[J].电脑知识与技术,2022,18(10):38-39.DOI:10.14004/j.cnki.ckt.2022.0744.

[2]唐蓉,周瑜平等.基于 RSA 算法特性的安全性研究[J].电子计工程,2023,31(04):164-168.DOI:10.14022/j.issn1674-6236.2023.04.034.

[3]王鑫淼,孙婷婷,马晶军.RSA 算法在网络数据传输中的研究进展[J].计算机科学,2023,50(S1):703-709.

[4]【【8086 汇编入门】《零基础入门学习汇编语言》】
https://www.bilibili.com/video/BV1Rs411c7HG?vd_source=b771feed7a3939c0a3130568bee8e248

[5]【十分钟学会编程的本质【收藏级】】

https://www.bilibili.com/video/BV1AF411s78P?vd_source=b771feed7a3939c0a3130568bee8e248

[6] 《8086/8088 汇编语言程序设计》 四川大学教材