**Introduction to Computer Science**

**Homework 6**

**沙行勉教授**

**Due Date: 星期二，Nov 20, 2018**

**Do not copy others, or you will get punished. If you allow other students to copy your report, you will be punished as well. In the homework, you will get more experience in programming. Good for you. 但是，对于编程题，我希望有基础的人能多帮助其他同学，帮助他们debug, 但是不要让他们抄袭。好吗？**

**PUT DOWN YOUR NAME and EMAIL ADDRESS IN YOUR REPORT.** The Filename of your report should be HOMEWORK N\_Student ID\_your name\_your Teacher’s last name (Sha or Zhang), where N is the homework number, for example, *1\_XiaoMing\_2018XXXX\_Sha*. Then send your report to the TA of your class. 注意，两班的TA不一样，千万不要送错了。**你可以用中文来回答问题。**

**请同学阅读书本的第三章前半章和我们的工具SEAL。这不是准备高考，你已经是大学生了，要自己学习知识，没人逼你用刷题的方式来学习了。当作有趣的知识来阅读。好吗？**

1. **（每题5 points）3.2.1, 3.2.2, 3.2.3, 3.2.4, 3.2.5, 3.2.6**

**3.2.1 mov R1,14h**

**store(1000),R1**

**3.2.2 将主存地址1200处的变量值读取到寄存器R2中**

**3.2.3 shiftl R3,R1,04h**

**3.2.4 shiftl R2,R1,02h**

**shiftl R4,R1,03h**

**add R3,R4,R2**

**3.2.5 load R1,(600)**

**load R2(604)**

**sub R1,R1,R2**

**store (600),R1**

**3.2.6 当用补码形式来表示二进制数时，对于正数，右移n位并补0之后的数是原来的数除以2的n次方后的商，而对于负数，右移n位并补1，对这个数进行取反并+1，所补的1全变成了0，也就是说，这种方法使得负数在移位时保证了计算的正确性，使得负数也能很好的进行除法运算，所以该指令的功能为：让带符号的二进制数进行准确的除法运算**

1. **（30 points）Let ID=your student ID. Then, let k=ID %2. So k can be 0 or 1. Based on the value of your ID, you get value of k. Please tell us your k value.**

**If k==0: 做习题3.3, 3.5, 3.7, 3.9, 3.11, 3.13   
If k==1: 做习题 3.2, 3.4, 3.6, 3.8, 3.10, 3.12  
k==1,所以做3.2,3.4,3.6,3.8,3.10,3.12**

**3.2:R2=R1+30=50,所以800处的结果为50**

**3.4：slt R4,R1,R2**

**beqz R4,L1**

**add R3,R1,R2**

**L1:**

**mov R3,R2**

3.6 **slt R4,R1,R2**

**beqz R4,L1**

**add R3,R1,R2**

**L1:**

**mov R3,R2**

**loop:**

**slt R5,R3,10**

**beqz R5,L2**

**add R3,R3,10**

**goto loop**

**L2:**

**3.8: while a<10:**

**a=a+b**

**b=b+1**

**3.10: while i<10:**

**a=a+b**

**b=b-1**

**i=i+1**

**(2):i=0,a=30,b=19,i=1**

**i=1,a=49,b=18,i=2**

**i=2,a=67,b=17,i=3**

**i=3,a=84,b=16,i=4**

**i=4,a=100,b=15,i=5**

**i=5,a=115,b=14,i=6**

**i=6,a=129,b=13,i=7**

**i=7,a=142,b=12,i=8**

**i=8,a=154,b=11,i=9**

**i=9,a=163,b=10,i=10**

**所以R2=163，R3=10**

**3.12：**

**a,b均是全局变量**

**a=b=30**

**print(a)#输出a=30**

**print(a)#输出a=30**

**所以没有局部变量，都是全局变量，输出结果为：**

**30**

**30**

**3.、（20 points）装上我们的SEAL（教学用的汇编语言模拟器）写出正确的汇编语言程序。输入：三个整数，用\_data 1,[a,b,c]表示，计算出最大数放入寄存器R1中，打印。请不要用函数调用。**

**请在注释中写出你的Python对应代码，把你的全部代码拷贝到word报告中，并将程序的运行结果截图粘贴到报告中。**

**#<python代码：计算最大数>**

**#L=[a,b,c]**

**#if L[0]<L[1]:**

**# max=L[1]**

**#else:**

**# max=L[0]**

**#if max<L[2]:**

**# max=L[2]**

**#else:**

**# max=max**

**#print(max)**

**#<seal汇编：计算最大数>**

**\_data 0,[17,13,6]**

**move R0,0**

**load R2,0(R0)**

**load R3,1(R0)**

**load R4,2(R0)**

**slt R5,R2,R3**

**beqz R5,L1**

**move R1,R3**

**goto L2**

**L1:**

**move R1,R2**

**L2:**

**slt R6,R1,R2**

**beqz R6,L3**

**move R1,R2**

**L3:**

**图片包含 屏幕截图

自动生成的说明\_pr R1**

**4、（20 points）请用SEAL写出和执行汇编语言程序。输入一个正整数，放在寄存器R0中，请用最简单的方式计算出它的二进制数有多少个1，将结果放在R1中，并打印。例如 move R0, 13, 结果是3。**

**请在注释中写出你的Python对应代码，把你的全部代码拷贝到word报告中，并将程序的运行结果截图粘贴到报告中。**

**#<python：计算十进制数的二进制形式有多少个1>**

**#R1=0；R0=13**

**#while R0!=0:**

**# R2=R0//2**

**# cha=R0-R2\*2**

**# if cha==1:**

**# n+=1**

**# R0=R2**

**#print(R1)**

**#<汇编语言：计算十进制的二进制形式有多少个1>**

**move R0,13**

**move R1,0**

**L1:**

**sle R3,R0,0**

**beqz R3,L2**

**goto L3**

**L2:**

**div R2,R0,2**

**mul R4,R2,2**

**sub R5,R0,R4**

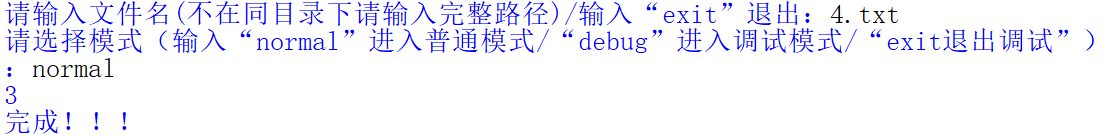
**move R0,R2**

**beqz R5,L1**

**add R1,R1,1**

**goto L1**

**L3:**

**\_pr R1**

**5、（20 points）请用SEAL写出和执行汇编语言程序。输入一个个字符串表示二进制数，以\_data的形式输入，此数组中的第一个数代表字符串的长度，然后是0或1的ASCII码：字符0是48，字符 1是49。例如，“101”是用\_data 0, [3, 49, 48, 49].是请用最简单的方式计算出它的二进制数有多少个”1”，将结果放在R1中，并打印。**

**请在注释中写出你的Python对应代码，把你的全部代码拷贝到word报告中，并将程序的运行结果截图粘贴到报告中。**

**#<python:计算有多少个1>**

**#n=101;L=[3,49,48,49]**

**#m=0**

**#i=1**

**#while i <len(L):**

**# if L[i]-49==0:**

**# m+=1**

**# i=i+1**

**#print(m)**

**\_data 0,[3,49,48,49]**

**move R0,1**

**load R2,(0)**

**move R1,0**

**L1:**

**sle R4,R0,R2**

**beqz R4,L2**

**load R5,0(R0)**

**sub R6,R5,49**

**add R0,R0,1**

**beqz R6,L3**

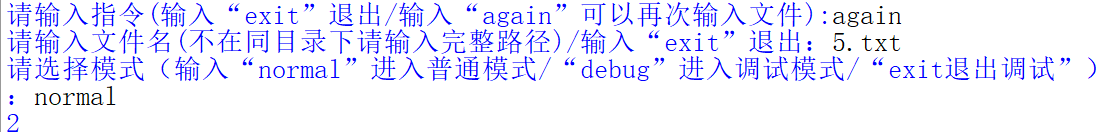
**sub R1,R1,1**

**L3:**

**add R1,R1,1**

**goto L1**

**L2:**

**\_pr R1**