HW3

姓名：王艺洲　　学号：PB21111722

题目 1. 假设寄存器 t_0 中初始状态下保存的值为 0x00002023。请回答下面的问题:

1. 对于指令 sub t2, t0, t1, 导致结果溢出的 t1 的值的范围？
2. 假设 PC（程序计数器）当前值为 0x0D000000, 则 jal 指令可以到达的地址范围是多少？如果是 blt 呢？

1.0x00002024~0xF0002023.

2.jal:0x0CE00000~0x0D200000.

blt:0x0CFFF000~0x0D001000.

题目 2. 本题目中所述的 int 整型变量都是 32 位的。

1. 将 $-a*2-(b+c)-(d+b+c)+200$ 转换为 RV32I 指令（a, b, c, d 均为 int 整型数值，且已经分别保留在寄存器 t0, t1, t2, t3 中。不考虑溢出问题）。请尝试使用尽可能少的寄存器和尽可能少的指令。
2. 将 $A[2*j]=B[i-8]$ 转换为 RV32I 指令，其中 A, B 为 int 整型数组。它们的基址分别保存在寄存器 a0, a1 中。 i, j 均为 int 整型变量，且已保存在寄存器 t0, t1 中（不考虑溢出以及非法访问问题，所有数据都已经四字节对齐）。

> 尽可能为你的指令添加注释，使得助教能够更好的理解你的代码。

1.

```
add t0, t0, t0      # t0=2a
sub t0, x0, t0      # t0=-2a
add t1, t1, t2      # t1=b+c
sub t0, t0, t1      # t0=-2a-(b+c)
sub t0, t0, t1      # t0=-2a-(b+c)-(b+c)
sub t0, t0, t3      # t0=-2a-(b+c)-(d+b+c)
addi t0, t0, 200    # t0=-2a-(b+c)-(d+b+c)-200
```

2.

```
addi t0, t0, -8     # t0=i-8
add t1, t1, t1      # t1=2j
slli t0, t0, 5      # t0=32*(i-8)
slli t1, t1, 5      # t1=32*(2*j)
add a0, a0, t1      # a0=A[2*j]
add a1, a1, t0      # a1=B[i-8]
lw t2, 0(a1)        # t2=B[i-8]
sw t2, 0(a0)        # A[2*j]=B[i-8]
```

3.

题目 3. 我们知道 RISC-V 存储是小端序的，即低地址存储低位，高地址存储高位。阅读如下代码：

```
lb  t1 ,  1(t0 );
sw  t1 ,  4(t0 );
```

初始条件下，t0 的内容为 0x2023, 地址 0x2023 的内容为 0x20881124。请问:

1. 该代码执行后，地址 0x2030 的内容是什么？
2. 如果 RISC-V 是大端序存储的，那么该代码执行后，地址 0x2030 的内容是什么？

1.0x00

2.0x88

4.

题目 4. 现在我们需要使用 RV32I 指令求解斐波那契数列的前 n 项，其中 n 为 int 整型变量，保存在内存地址 place 中。斐波那契数列的第一项和第二项分别保存在内存地址 first 和 second 中。请根据以上信息编写 RV32I 指令，将从第一项开始的结果依次保存在从内存地址 save 开始的连续内存中。

```
        la t0, place        # t0保存n的地址
        lw t1, 0(t0)        # t1保存n
        la t2, first        # t2保存第一项的地址
        la t3, second       # t3保存第二项的地址
        la t4, save         # t4保存第一项开始的地址
        sw t2, 0(t4)        # 保存第一项
        addi t4, t4, 4      # t4指向下一个位置
        addi t1, t1, -2     # n=n-1
        beqz t1, done       # 若n等于0，则结束
        sw t3, 0(t4)        # 保存第二项
        addi t4, t4, 4      # t4指向下一个位置
        addi t1, t1, -2     # n=n-1
        beqz t1, done       # 若n等于0，则结束
loop    lw t2, -8(t4)       # t2保存第一个数
        lw t3, -4(t4)       # t3保存第一个数
```

```
        add t0, t2, t3      # t0=t2+t3
        sw t0, 0(t4)        # 保存两个数的和
        addi t2, t2, 4
        addi t3, t3, 4      #两项位置更新
        addi t4, t4, 4      #保存位置更新
        addi t1 t1 -1       #n=n-1
        beez t1, loop       # 若n等于0，则结束
done    li v0, 10           #结束
```