Question 3 (10 points):

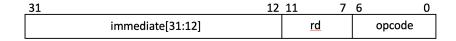


Figure 1: U-Type Format. Used for lui instructions in RISC-V.

In a lab assignment for CMPUT 229 you are building a simulator for a RISC-V processor. One of the data structures in this simulator is a table that contains the value of each of the 32 registers in the simulated RISC-V processor. This table is called regTable. Each entry in regTable is a 32-bit word that corresponds to the current value stored in that register in the simulated RISC-V processor. The position xi of regTable contains the value of register xi in the simulated processor. For instance the value stored in regTable[0] should always be zero.

In this question you need to write the RISC-V code for the function luiUpdate for this simulator. luiUpdate has two arguments:

- a0: binary representation of a lui instruction
- a1: address of first position of regTable

luiUpdate changes the contents of regTable according to the semantics of the instruction lui. The instruction lui uses the U-Type format shown in Figure 1

```
1
  # luiUpdate:
2 # arguments:
3 #
        a0: binary representation of a lui instruction
4
  #
        a1: address of first position of regTable
5
  luiUpdate:
6
       slli t0, a0, 20
7
       srli t0, t0, 27 # t0 <- rd
8
            t1, t0, 2 # t1 <- 4*rd
       add t2, a1, t1 # t2 <- Address(RegTable[rd])
9
       srli t3, a0, 12 #
10
       slli t3, t3, 12 # t3 <- value to store in register
11
            t3, 0(t2) # regTable[rd] <- value
12
13
       jal
            zero, ra, 0
```

Figure 2: MIPS Code for luiUpdate