HW4/LaypanovAdamey/224

Task1

0x6c6c6337=> 01101100011011000110001100110111
=> lui t1, 444102

0x54830313=> 01010100100000110000001100010011
=> addi t1, t1, 1352

0x412033b7=>01000001001000000011001110110111
=> lui t2, 266755

0xc6f38393=>11000110111100111000001110010011
=> addi t2, t2, -913

0x00535e37=>00000000010100110101111000110111
=> lui t3, 1333

0xf43e0e13=>11110100001111100000111000010011
=> addi t3, t3, -189

0x10010437=>00010000000000010000010000110111
=> lui s0, 65552

0x00040413=>00000000000001000000010000010011
=>addi s0, s0,

0x00642023=>00000000011001000010000000100011
=> sw t1, 0(s0)

0x00742223=> 00000000011101000010001000100011
=> sw t2, 4(s0)

0x01c42423=>00000001110001000010010000100011
=> sw t3, 8(s0)

0x00400893=>00000000010000000000100010010011
=> addi a7, zero, 4

0x00800533=>00000000100000000000010100110011
=> add a0, zero, s0

0x00000073=>00000000000000000000000001110011
=> ecall

```
lui t1, 444102
addi t1, t1, 1352
lui t2, 266755
addi t2, t2, -913
lui t3, 1333
addi t3, t3, -189
lui s0, 65552
addi s0, s0, 0
sw t1, 0(s0)
sw t2, 4(s0)
sw t3, 8(s0)
addi a7, zero, 4
add a0, zero, s0
ecall
```

Output:
Hello, world

короче, я не вижу другого сокращения кроме как просто убрать строку
addi s0, s0,

Task 2

0x00500893 => 0101 00000 000 10001 0010011   addi a7, zero, 5

0x00000073 => 01110011   ecall

0x00a00333 => 1010 00000 000 00110 0110011   add t1, zero, a0

0x01f55293 => 000 11111 01010 101 00101 0010011   srli t0, a0, 31

0x00000073 => 01110011   ecall

```
0x00a00eb3  => 1010 00000 000 11101 0110011    add t4, zero, a0
0x01f55e13 => 000 11111 01010 101 11100 0010011    srli t3, a0, 31
0x06030263 =>  011 00000 00110 000 00100 1100011    beq t1, zero, address_1
0x060e8063 => 011 00000 11101 000 00000 1100011    beq t4, zero, address_2
0x00028663 => 00101 000 01100 1100011    beq t0, zero, addres_3
0xfff34313 => 111111111111 00110 100 00110 0010011    xori t1, t1, -1
0x00130313 => 0001 00110 000 00110 0010011    addi t1, t1, 1
0x000e0663 => 11100 000 01100 1100011    beq t3, zero, address_4
0xfffece93 => 111111111111 11101 100 11101 0010011    xori t4, t4, -1
0x001e8e93 => 0001 11101 000 11101 0010011    addi t4, t4, 1
0x01d35863 => 000 11101 00110 101 10000 1100011    bge t1, t4, address_5
0x006eceb3 => 0110 11101 100 11101 0110011    xor t4, t4, t1
0x006ec333 => 0110 11101 100 00110 0110011    xor t1, t4, t1
0x006eceb3 => 0110 11101 100 11101 0110011    xor t4, t4, t1
0x000003b3 => 00111 0110011    add t2, zero, zero
0x006383b3  =>  0110 00111 000 00111 0110011    add t2, t2, t1
0xfffe8e93 => 111111111111 11101 000 11101 0010011    addi t4, t4, -1
0xffd04ce3 => 1111111 11101 00000 100 11001 1100011 blt zero, t4, adress_6
0x01c2cfb3 => 000 11100 00101 100 11111 0110011    xor t6, t0, t3
0x000f8663 => 11111 000 01100 1100011 beq t6, zero, address_7
0xfff3c393 =>111111111111 00111 100 00111 0010011    xori t2, t2, -1
0x00138393  =>  0001 00111 000 00111 0010011    addi t2, t2, 1
0x00100893  =>  0001 00000 000 10001 0010011    addi a7, zero, 1
0x00700533  =>  0111 00000 000 01010 0110011    add a0, zero, t2
0x00000073  =>  01110011    ecall
0x00a00893  => 1010 00000 000 10001 0010011    addi a7, zero, 10
0x00000073  =>  01110011    ecall
```

```
0x00100893  =>  0001 00000 000 10001 0010011    addi a7, zero, 1
0x00000533 =>  01010 0110011    add a0, zero, zero
0x00000073 =>  01110011    ecall
0x00a00893  => 0001 00000 000 10001 0010011    addi a7, zero, 10
0x00000073 =>  01110011    ecall
```

address_1 has imm = 100
address_2 has imm = 96
address_3 has imm = 12
address_4 has imm = 12
address_5 has imm = 16
address_6 has imm = -8
address 7 has imm = 12

FULL CODE:

```
addi a7, zero, 5
ecall
add t1, zero, a0
srli t0, a0, 31
ecall
add t4, zero, a0
srli t3, a0, 31
beq t1, zero, add1
beq t4, zero, add2
beq t0, zero, add3
xori t1, t1, -1
addi t1, t1, 1
add3:
beq t3, zero, add4
xori t4, t4, -1
addi t4, t4, 1
add4:
bge t1, t4, add5
xor t4, t4, t1
xor t1, t4, t1
xor t4, t4, t1
add5:
add t2, zero, zero
```

```
add6:
add t2, t2, t1
addi t4, t4, -1
blt zero, t4, add6
xor t6, t0, t3
beq t6, zero, add7
xori t2, t2, -1
addi t2, t2, 1
add7:
addi a7, zero, 1
add a0, zero, t2
ecall
addi a7, zero, 10
ecall
add1:
add2:
addi a7, zero, 1
add a0, zero, zero
ecall
addi a7, zero, 1
ecall
```

“геморный” алгоритм просто умножить 2 числа

Сокращенная программа

```
addi a7, zero, 5 -> введем число после ecall на след строке
ecall
add t0, zero, a0 -> t0 = a0
ecall - опять вызов ввода
add t1, zero, a0 -> t1 = a0
mul t2, t0, t1 -> t2=a*b
add a0, zero, t2 -> a0 = t2
addi a7, zero, 1-> вывод
ecall
```

Другая версия:

```
li a7, 5
ecall
mv t0, a0
```

```
ecall
mv t1, a0
mul t2, t0, t1
mv a0, t2
li a7, 1
ecall
```

(на 1 строчку больше, но более привычно мне)

btw, можно сократить

```
li a7, 5
ecall
mv t0, a0
ecall
mul a0, t0, a0
li a7, 1
ecall
```