עדיאל בן משה 208969378

Q1

```
Zelensky: # address(Zelensky) = 0xaa00150c = 10101010 00000000 00010101 00001100
subu $t0, $zero, $s3
# 000000 00000 10011 01000 00000 100011
# 00000000000100110100000000100011
# 4-base: 00000103 10000203
# 0x00134023
# op=0x00 rs=0x00 rt=0x13 rd=0x08 shamt=0x00 funct=0x23
# op=0 rs=0 rt=19 rd=8 shamt=0 funct=35
# op=000000 rs=00000 rt=10011 rd=01000 shamt=00000 funct=100011

"asdasd"
bgtz $a3, Zelensky
# 000111 00111 00000 1111111111111110
# 00011100111000001111111111111110
# 0x1CE0FFFE
# 4-base: 01303200 33333332
# op=000111 rs=00111 rt=00000 imm=1111111111111110
# op=0x07 rs=0x07 rt=0x00 imm=0xFFFE
# op=07 rs=7 rt=0 imm=-2

mfhi $t3
# 000000 00000 00000 01011 00000 010000
# 00000000000000000101100000010000
# 0x00005810
# 4-base: 00000000 11200100
# op=000000 rs=00000 rt=00000 rd=01011 shamt=00000 funct=010000
# op=0x00 rs=0x00 rt=0x00 rd=0x0B shamt=0x00 funct=0x10
# op=0 rs=0 rt=0 rd=11 shamt=0 funct=16

lhu $t2, -12($s6)
# 100101 10110 01010 1111111111110100
# 10010110110010101111111111110100
# 0x96CAFFF4
# 4-base: 21123022 33333310
# op=100101 rs=10110 rt=01010 imm=1111111111110100
# op=0x25 rs=0x16 rt=0x0A imm=0xFFF4
# op=37 rs=22 rt=10 imm=-12

jal Zelensky
# 000011 10100000000000010101000011
# 00001110100000000000010101000011
# 0x0E800543
# 4-base: 00322000 00111003
# op=000011 address=10100000000000010101000011
# op=0x03 address=0x02800543
# op=3 address=-10
```

Q2

.

	Psuedoinstruction
`$t1 = $t2`	`move $t1, $t2`
`$t0 = 0`	`clear $t0`
`if ($t1 = small) go to L`	`beq $t1, small, L`
`if ($t1 = big) go to L`	`beq $t1, big, L`
`$t1 = small`	`li $t1, small`
`$t2 = big`	`li $t2, big`
`if ($t3 <= $t5) go to L`	`ble $t3, $t5, L`
`if ($t4 > $t6) go to L`	`bgt $t4, $t6, L`
`if ($t5 >= $t3) go to L`	`bge $t5, $t3, L`
`$t0 = $t2 + big`	`addi $t0, $t2, big`
`$t5 = memory[$t2 + big]`	`lw $t5, big($t2)`

```
# pseudo
move $t1, $t2
# native
add $t1, $t2, $zero
# ----------------
# pseudo
clear $t0
# native
add $t0, $zero, $zero
# ----------------
# pseudo
beq $t1, [small], L
# native
addi $at, $zero, [small]
beq $t1, $at, L
# ----------------
# pseudo
beq $t1, [big], L
# native
lui $at, [big_upper]
ori $at, $at, [big_lower]
beq $t1, $at, L
# ----------------
# pseudo
li $t1, [small]
# native
addi $t1, $zero, [small]
# ----------------
# pseudo
li $t2, [big]
# native
lui $t2, [big_upper]
ori $t2, $t2, [big_lower]
# ----------------
# pseudo
ble $t3, $t5, L
```

```
# native
slt $at, $t5, $t3
beq $at, $zero, L
# ----------------
# pseudo
bgt $t4, $t6, L
# native
slt $at, $t6, $t4
bne $at, $zero, L
# ----------------
# pseudo
bge $t5, $t3, L
# native
slt $at, $t5, $t3
beq $at, $zero, L
# ----------------
# pseudo
addi $t0, $t2, [big]
# native
lui $at, [big_upper]
ori $at, $at, [big_lower]
add $t0, $t2, $at
# ----------------
# pseudo
lw $t5, [big]($t2) # $t5 = memory[$t2 + big]
# native
lui $at, [big_upper]
addu $at, $at, $t2
lw $t5, [big_lower]($at)
```

Q3

Text Segment

Bkpt	Address	Code	Basic	Source
				Source code line
	0x00400078	0x002f0821	addu $1,$1,$15	
	0x0040007c	0x802b0000	lb $11,0x00000000($1)	
	0x00400080	0x000b2021	addu $4,$0,$11	70: move $a0,$t3
	0x00400084	0x2402000b	addiu $2,$0,0x0000000b	71: li $v0,11
	0x00400088	0x0000000c	syscall	72: syscall
	0x0040008c	0x20010001	addi $1,$0,0x00000001	73: subi $t7,$t7,1
	0x00400090	0x01e17822	sub $15,$15,$1	
	0x00400094	0x15e0fff7	bne $15,$0,0xfffffff7	74: bne $t7,$zero,reverse
	0x00400098	0x2402000a	addiu $2,$0,0x0000000a	80: li $v0,10
	0x0040009c	0x0000000c	syscall	81: syscall

MARS Text Segment

```
# text segment:
# start address: 0x00400078
---
0x002f0821              addu $1, $1, $15
0x802b0000              lb $11, 0x00000000($1)
0x000b2021              addu $4, $0, $11                    70:  move $a0, $t3
0x2402000b              addiu $2, $0, 0x0000000b    71: li $v0, 11
0x0000000c              syscall                                 72: syscall
0x20010001              addi $1, $0, 0x00000001     73: subi $t7, $t7, 1
0x01e17822              sub $15, $15, $1
0x15e0fff7              bne $15, $0, 0xfffffff7     74: bne $t7, $zero, reverse
0x2402000a              addiu $2, $0, 0x0000000a    80: li $v0, 10
0x0000000c              syscall                                 81: syscall
```

- (A)
  - what can we say about the instruction in line 73 in the source code `subi`?
    - it's not a valid instruction
  - will the program pass compilation?
    - no
  - is it possible to perform an effective translation of the MARS translation as it appears in the Basic column?
    - using the `addi $t7, $t7, -1` instruction
- (B) can we know what will happen following the `syscall` instruction in line 72? explain!
  - **answer**: the program will print the character that its ASCII code is in register `$a0` which is the value of `$t3`
- (C) can we know what will happen following the syscall instruction in line 81? explain!
  - **answer**: the program will terminate (syscall `10`)
- (D) based on the data in the segment code, what is the address of the label `reverse`? explain!
  - `0xfffffff7 = -9`
  - `0x00400094` is the address of `bne ...`
  - `0x00400094 + 4 + (-9 * 4)`
  - `0x00400094 + 4 - 0x24 = 0x00400074` is the address of `reverse`