

שאלה 1: (השימוש בצבעים נועד להדגיש את שם השדה בפקודה למשל immediate בצהוב funct בכחול וכו')

	decimal divided to command parts							
κ'		opcode	address					
					rd	shamt	funct	
	Type		rs	rt	immediate			
	I	32	10	8	-3			lb \$t0,-3(\$t2)
	I	4	8	0	3			beq \$t0,\$zero,strEnd
	R	0	25	8	25	0	32	add \$t9,\$t9,\$t0
	I	8	10	10	-12			addi \$t2,\$t2,-12
	J	2	1048640					j nextCh
ρ'	hexa divided to command parts							
		opcode	address					
					rd	shamt	funct	
	Type		rs	rt	immediate			
	I	0x20	0xa	0x8	0xfffd			lb \$t0,-3(\$t2)
	I	0x4	0x8	0x0	0x3			beq \$t0,\$zero,strEnd
	R	0x0	0x19	0x8	0x19	0x0	0x20	add \$t9,\$t9,\$t0
	I	0x8	0xa	0xa	0xff4			addi \$t2,\$t2,-12
J	0x2	0x100040					j nextCh	
	binary divided to command parts							
λ'		opcode	address					
					rd	shamt	funct	
	Type		rs	rt	immediate			
	I	0b100000	0b01010	0b01000	0b1111111111111101			lb \$t0,-3(\$t2)
	I	0b000100	0b01000	0b00000	0b0000000000000011			beq \$t0,\$zero,strEnd
	R	0b000000	0b11001	0b01000	0b11001	0b00000	0b100000	add \$t9,\$t9,\$t0
	I	0b001000	0b01010	0b01010	0b1111111111110100			addi \$t2,\$t2,-12
	J	0b000010	0b000000010000000000001000000					j nextCh
Τ'	binary							
	1000 0001 0100 1000 1111 1111 1111 1101							lb \$t0,-3(\$t2)
	0001 0001 0000 0000 0000 0000 0000 0011							beq \$t0,\$zero,strEnd
	0000 0011 0010 1000 1100 1000 0010 0000							add \$t9,\$t9,\$t0
	0010 0001 0100 1010 1111 1111 1111 0100							addi \$t2,\$t2,-12
	0000 1000 0001 0000 0000 0000 0100 0000							j nextCh

h'	hexadecimal							
	0x8148fffd							lb \$t0,-3(\$t2)
	0x11000003							beq \$t0,\$zero,strEnd
	0x0328c820							add \$t9,\$t9,\$t0
	0x214afff4							addi \$t2,\$t2,-12
	0x08100000							j nextCh

שאלה 2: `abs $t2,$t3`

את הפעולה `abs` ניתן לתרגם לשלוש פקודות מסט הפקודות המקורי:

`#shift right arithmetic 31 bit to make the sign bit spread to all other bits and store at $at`

`#this means that $at is 0xffffffff or 0x00000000 depends on the sign`

`sra $at $t3,31`

`#xor $at with the original number (this inverts all the numbers if $at is 0xffffffff and does`

`#nothing if $at is 0x00000000) inverting all bit in two's complement is  $\text{invert}(x) = (-x) - 1$`

`xor $t2,$t3,$at`

`#subtract $at from $t2: if $at is 0xffffffff=-1 then the result will be  $(-x) - 1 - (-1) = -x$`

`#$at is 0xffffffff only if x is negative meaning that  $\text{abs}(x) = \text{invert}(x) - 1 = (-x) - 1 - (-1) = -x$  ;(if x is`

`#negative) and  $\text{abs}(x) = (x \text{ xor } 0) - (0) = x$  ;(if x is positive as intended)`

`sub $t2,$t2,$at #subtract xor answer with the spreaded signed bit a $at`

שאלה 2: `min $t1,$t2,$t3`

`slt $at,$t2,$t3 # $at is set to 1 if $t2 is smaller then $t3`

`beq $at,$zero,option1 #if $at is 0 (meaning that $t3 is smaller) go to option1`

`add $t1,$t2,$zero #else $at is 1 meaning that $t2 is smaller) move $t2 to $t1`

`j continue #jump over the branch`

`option1:add $t1,$t3,$zero # $t3 is smaller so move $t3 to $t1`

`continue:`

שאלה 3:

א. התוכנית תעבור הידור משום שהפקודה `subi` נמצאת בסט הפסאודו פקודות של המיפס (או לפחות של סימולטור המרס הנוכחי)

התרגום היעיל ביותר הוא פשוט שימוש ב `addi` עם סימן מינוס:

```
addi $t7,$t7,-1
```

ב. בעקבות הפקודה `syscall` בשורה 72 יודפס `char` שמקודד ב: `t3$`

ג. בעקבות הפקודה תסתיים התוכנית.

ד. הערך היחסי בפקודת הענף (שדה ה `immediate`) הינו `0xffffffff7` שזה 9-. הכפ בעת הפקודה כבר התקדם ל `0x00400098`. נוריד מ `0x00400098` את המספר  $4 \times 9$  (9 כפול ארבעה בתים) ונקבל `0x00400074` הכתובת של `reverse`.