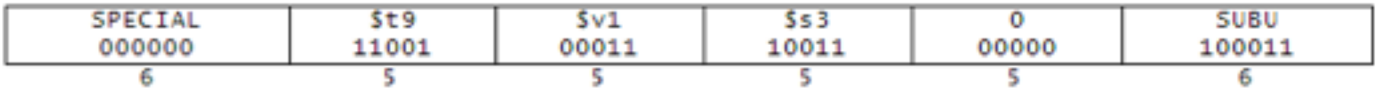
Q1.

```
li $v0,5
syscall
move $s0, $v0   # move the value to $s0 from $v0
blt $s0,$zero,next  # if the number is negative,execute next
move $t0,$s0   # move to $t0
next:
sub $t0,$zero,$s0  # if the number is negative subtract 0 to get the absolute value
```

Q2.

A.

subu $s3, $t9, $v1

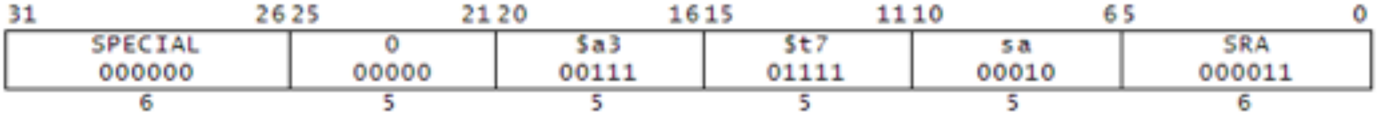


SPECIAL 000000	$t9 11001	$v1 00011	$s3 10011	0 00000	SUBU 100011
6	5	5	5	5	6

000000|11001|00011|10011|00000|100011

Answer 0x03239823

sra $t7,$a3,2



31 26	25 21	20 16	15 11	10 6	5 0
SPECIAL 000000	0 00000	$a3 00111	$t7 01111	sa 00010	SRA 000011
6	5	5	5	5	6

000000|00000|00111|01111|00010|000011

Answer 0x00077883

B.

0x144C0012

00010100010011000000000000010010

BNE $v0, $t4, 0x0012(18)

C.

la is an instruction that takes an address and puts it into a register

li is an instruction that loads an immediate value into a register

These instructions can’t be regular instructions because they require more than 32 bits.

The real instruction for li $t0, 0 is

```
lui $t0, 0
ori $t0, $t0, 0
```

Q3.

```
##############################################
##                                                          .text
#                              #                            .globl __start
#          text segment              #                  __start:
#                              #                            addi $s0, $0, 5
##############################################              addi $s1, $0, 25
##
```

```
    la $a0, promptS         #prompt for S
    li $v0, 4
    syscall

    li $v0, 5       #read S
    syscall

    move $t0, $v0           #store S into t0

    la $a0, promptA         #prompt for A
    li $v0, 4
    syscall

    li $v0, 5       #read A
    syscall

    beq $t0, $s0, casefive   #if S == 5
    beq $t0, $s1, casetwentyfive  #if S == 25
    sll $s2, $v0, 1      #shift left once to multiply by 2,
sign is still preserved

    j answer       #output answer

casefive:     addi $s2, $v0, 1   #add 1 to A

    j answer       #output answer

casetwentyfive:
    addi $s2, $v0, -1  #sub 1 from A
                   #output answer

answer:  la $a0, ans          #prep for answer
    li $v0, 4
    syscall

    move $a0, $s2    #move integer answer for
output (always in s2)
    li $v0, 1
    syscall

    la $a0, endl         #make the output pretty
    li $v0, 4
    syscall

    li $v0, 10
    syscall         #exit

#############################################
##
#                                      #
#            data segment                   #
#                                      #
#############################################
##
```

```
    .data
promptS:    .asciiz "Please enter an integer for
variable S.\n"
promptA:    .asciiz "Please enter an integer for
variable A.\n"
ans:      .asciiz "The value of A is now: "
endl:           .asciiz "\n"
```