1.

.text

.globl main

main:

la $s1, 0x80000001

la $s2, 0x80000001

add $s, $s1, $s2 # $s3 <- $s1 + $s2 (overflow)

addu $s4, $s1, $s2 # $s4 <- $s1 + $s2 (done)

# I-type uses a 16-bit constant value. (e.g 0x1234)

la $s5, 0xefffffff

addiu $s6, $s5, 0x0fff # $s6 <- $s5 + 0x0fff

li $v0, 10

syscall

이유: 맨 앞의 4개의 bit에서 1111 + 1111 을 하면 올림값때문에 4자리수를 넘어가는 값이 나오므로 오버플러가 된다.

2.

.text

.globl main

main:

la $s1, 0x90000001

la $s2, 0x70000001

sub $s3, $s1, $s2 # $s3 <- $s1 - $s2 (overflow)

subu $s4, $s1, $s2 # $s4 <- $s1 - $s2

li $v0, 10

syscall

이유: 부호를 확인하는 sub경우 1001(9) 과 0111(7)을 빼주어야하는데 1001을 부호확인하면 -1 이고 0111을 2의 보수로 취하면 1001(-7)이되고 서로 더해주면 오버플러워 되는 값이 발생한다.

3.

.text

.globl main

main:

la $s0, 0xffffffff # 0xffffffff == -1

sll $s1, $s0, 4 # $s1 <- $s0 << 4

srl $s2, $s0, 4 # $s2 <- $s0 >>> 4

sra $s3, $s0, 4 # $s3 <- $s0 >> 4

li $v0, 10

syscall