

BÁO CÁO TUẦN 4

Bài 1:

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

.data

x: .word 0x80000000

y: .word 0x7fffffff

isOverflow: .word 0

.text

start:

```
lw    $s1,x           #Load value x
lw    $s2,y           #Load value y
la    $s4,isOverflow  #Load address isOverflow
li    $t0,0           #No Overflow is default status
addu  $s3,$s1,$s2     # s3 = s1 + s2
xor    $t1,$s1,$s2    #Test if $s1 and $s2 have same sign
```

```
bltz   $t1,EXIT        #If not, exit
slt    $t2,$s3,$s1
bltz   $s1, NEGATIVE   #Test if $s1 and $s2 is nagative?
beq    $t2,$zero,EXIT  #$s1 and $s2 are positive
#if $s3 > $s1 then the result is not overflow
j      OVERFLOW
```

NEGATIVE:

```
bne    $t2,$zero,EXIT  #$s1 and $s2 aer negative
#if $s3 < $s1 then the result is not overflow
```

OVERFLOW:

```
li    $t0,1           #The result is overflow
```

EXIT:

```
sw    $t0,0($s4)
```

TH1: x, y trái dấu

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:          .word    0x80000000
y:          .word    0x7fffffff
isOverflow: .word    0

.text
start:

    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2     # s3 = s1 + s2
    xor     $t1,$s1,$s2     #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT        #If not, exit
    slt     $t2,$s3,$s1     #Test if $s1 and $s2 is negative?
    bltz    $s1, NEGATIVE   #Test if $s1 and $s2 is negative?
    beq     $t2,$zero,EXIT  #If $s3 > $s1 then the result is not overflow
    j       OVERFLOW

NEGATIVE:
    bne     $t2,$zero,EXIT  #If $s3 < $s1 then the result is not overflow

OVERFLOW:
    li      $t0,1          #The result is overflow

EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment						Labels	
Bkpt	Address	Code	Basic	Source		Label	Address
<input type="checkbox"/>	4194304	0x3c011001	lui \$1,4097	8: lw \$s1,x	#Load value x	start	4194304
<input type="checkbox"/>	4194308	0x8c310000	lw \$17,0(\$1)	9: lw \$s2,y	#Load value y	NEGATIVE	4194360
<input type="checkbox"/>	4194312	0x3c011001	lui \$1,4097	10: la \$s4,isOverflow	#Load address isOverflow	OVERFLOW	4194364
<input type="checkbox"/>	4194316	0x8c320004	lw \$18,4(\$1)	11: li \$t0,0	#No Overflow is default status	EXIT	4194368
<input type="checkbox"/>	4194320	0x3c011001	lui \$1,4097	12: addu \$s3,\$s1,\$s2	# s3 = s1 + s2	x	268500992
<input type="checkbox"/>	4194324	0x34340008	ori \$20,\$1,8	13: xor \$t1,\$s1,\$s2	#Test if \$s1 and \$s2 have same sign	y	268500996
<input type="checkbox"/>	4194328	0x24080000	addiu \$6,\$0,0	14: bltz \$t1,EXIT	#If not, exit	isOverflow	268501000
<input type="checkbox"/>	4194332	0x02329821	addu \$19,\$17,\$18	15: slt \$t2,\$s3,\$s1	#Test if \$s1 and \$s2 is negative?		
<input type="checkbox"/>	4194336	0x02324826	xor \$9,\$17,\$18	16: bltz \$s1, NEGATIVE	#Test if \$s1 and \$s2 is negative?		
<input type="checkbox"/>	4194340	0x05200006	bltz \$9,\$6	17: beq \$t2,\$zero,EXIT	#If \$s3 > \$s1 then the result is not overflow		
<input type="checkbox"/>	4194344	0x0271502a	bltz \$10,\$19,\$17	18: j OVERFLOW			
<input type="checkbox"/>	4194348	0x06200002	bltz \$17,\$2	19: bne \$t2,\$zero,EXIT	#If \$s3 < \$s1 then the result is not overflow		
<input type="checkbox"/>	4194352	0x11400003	beq \$10,\$0,\$3	20: li \$t0,1	#The result is overflow		
<input type="checkbox"/>	4194356	0x0810000f	j 4194364	21: sw \$t0,0(\$s4)			
<input type="checkbox"/>	4194360	0x15400001	bne \$10,\$0,\$1				
<input type="checkbox"/>	4194364	0x24080001	addiu \$6,\$0,1				
<input type="checkbox"/>	4194368	0x24080001	addiu \$6,\$0,1				
<input type="checkbox"/>	4194372	0x24080001	addiu \$6,\$0,1				

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2147483648	2147483647	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

\$s1	17	-2147483648
\$s2	18	2147483647
\$s3	19	-1
\$s4	20	268501000

TH2: x, y cùng dương và x+y không tràn

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:          .word    123456
y:          .word    123457
isOverflow: .word    0
.text
start:
    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2     # s3 = s1 + s2
    xor     $t1,$s1,$s2     #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT        #If not, exit
    slt     $t2,$s3,$s1     #Test if $s1 and $s2 is nagative?
    bltz    $s1, NEGATIVE   #Test if $s1 and $s2 is nagative?
    beq     $t2,$zero,EXIT  #If $s3 > $s1 then the result is not overflow
    j       OVERFLOW
NEGATIVE:
    bne     $t2,$zero,EXIT  #If $s3 < $s1 then the result is not overflow
    j       OVERFLOW
OVERFLOW:
    li      $t0,1           #The result is overflow
EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment						Labels	
Bkpt	Address	Code	Basic	Source		Label	Address
	4194304	0x3c011001	lui \$1,4097	8: lw \$s1,x	#Load value x	start	4194304
	4194308	0x8c310000	lw \$17,0(\$1)			NEGATIVE	4194360
	4194312	0x3c011001	lui \$1,4097	9: lw \$s2,y	#Load value y	OVERFLOW	4194364
	4194316	0x8c320004	lw \$19,4(\$1)			EXIT	4194368
	4194320	0x3c011001	lui \$1,4097	10: la \$s4,isOverflow	#Load address isOverflow	x	268500992
	4194324	0x34340008	ori \$20,\$1,8			y	268500996
	4194328	0x24080000	addiu \$8,\$0,0	11: li \$t0,0	#No Overflow is default status	isOverflow	268501000
	4194332	0x02329821	addu \$19,\$17,\$18	12: addu \$s3,\$s1,\$s2	# s3 = s1 + s2		
	4194336	0x02324826	xor \$9,\$17,\$18	13: xor \$t1,\$s1,\$s2	#Test if \$s1 and \$s2 have same sign		
	4194340	0x05200006	bltz \$9,6	15: bltz \$t1,EXIT	#If not, exit		
	4194344	0x0271502a	slt \$10,\$19,\$17	16: slt \$t2,\$s3,\$s1			
	4194348	0x06200002	bltz \$17,2	17: bltz \$s1,NEGATIVE	#Test if \$s1 and \$s2 is nagative?		
	4194352	0x11400003	beq \$10,\$0,3	18: beq \$t2,\$zero,EXIT	#s1 and \$s2 are positive		
	4194356	0x010000f9	4194364	20: j OVERFLOW	#s1 and \$s2 aer negative		
	4194360	0x19400001	bne \$10,\$0,1	22: bne \$t2,\$zero,EXIT	#The result is overflow		
	4194364	0x24080001	addiu \$8,\$0,1	25: li \$t0,1			
	4194368	0xaa880000	sw \$8,0(\$20)	27: sw \$r0,0(\$s4)			

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	123456	123457	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

\$s1	17	123456
\$s2	18	123457
\$s3	19	246913
\$s4	20	268501000

TH3: x, y cùng âm và x+y không tràn

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:                .word    -123456
y:                .word    -923457
isOverflow:       .word    0

.text
start:

    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2     # s3 = s1 + s2
    xor     $t1,$s1,$s2     #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT        #If not, exit
    slt     $t2,$s3,$s1     #Test if $s1 and $s2 is negative?
    bltz    $s1, NEGATIVE   #Test if $s1 and $s2 is negative?
    beq     $t2,$zero,EXIT  #If $s1 and $s2 are positive
                                #if $s3 > $s1 then the result is not overflow
    j       OVERFLOW

NEGATIVE:
    bne     $t2,$zero,EXIT  #If $s1 and $s2 aer negative
                                #if $s3 < $s1 then the result is not overflow

OVERFLOW:
    li      $t0,1           #The result is overflow

EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment

Bkpt	Address	Code	Basic	Source
<input type="checkbox"/>	4194304	0x3c011001lui \$1,4097	8:	lw \$s1,x #Load value x
<input type="checkbox"/>	4194308	0x3c310000lw \$17,0(\$1)		
<input type="checkbox"/>	4194312	0x3c011001lui \$1,4097	9:	lw \$s2,y #Load value y
<input type="checkbox"/>	4194316	0x3c320004lw \$18,4(\$1)		
<input type="checkbox"/>	4194320	0x3c011001lui \$1,4097	10:	la \$s4,isOverflow #Load address isOverflow
<input type="checkbox"/>	4194324	0x34340008ori \$20,\$1,8		
<input type="checkbox"/>	4194328	0x24080000addiu \$8,\$0,0	11:	li \$t0,0 #No Overflow is default status
<input type="checkbox"/>	4194332	0x03329921addu \$19,\$17,\$18	12:	addu \$s3,\$s1,\$s2 # s3 = s1 + s2
<input type="checkbox"/>	4194336	0x03324826xor \$9,\$17,\$18	13:	xor \$t1,\$s1,\$s2 #Test if \$s1 and \$s2 have same sign
<input type="checkbox"/>	4194340	0x05200006bltz \$9,6	15:	bltz \$t1,EXIT #If not, exit
<input type="checkbox"/>	4194344	0x02715026slti \$10,\$19,\$17	16:	slt \$t2,\$s3,\$s1
<input type="checkbox"/>	4194348	0x06200002bltz \$17,2	17:	bltz \$s1, NEGATIVE #Test if \$s1 and \$s2 is negative?
<input type="checkbox"/>	4194352	0x11400003beq \$10,\$0,3	18:	beq \$t2,\$zero,EXIT #s1 and s2 are positive
<input type="checkbox"/>	4194356	0x0810000fj 4194364	20:	j OVERFLOW
<input type="checkbox"/>	4194360	0x15400001bne \$10,\$0,1	22:	bne \$t2,\$zero,EXIT #s1 and s2 are negative
<input type="checkbox"/>	4194364	0x24080001addiu \$8,\$0,1	25:	li \$t0,1 #The result is overflow
<input type="checkbox"/>	4194368	0x24080001addiu \$8,\$0,1	27:	sw \$t0,0(\$s4)

Labels

Label	Address
as1w4.asm	
start	4194304
NEGATIVE	4194360
OVERFLOW	4194364
EXIT	4194368
x	268500992
y	268500996
isOverflow	268501000

☒ Data☒ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-123456	-923457	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

\$s1	17	-123456
\$s2	18	-923457
\$s3	19	-1046913
\$s4	20	268501000

TH4: x, y cùng dương và x+y tràn

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:          .word    0x7fffffff
y:          .word    1
isOverflow: .word    0
.text
start:
    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2     # s3 = s1 + s2
    xor     $t1,$s1,$s2     #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT        #If not, exit
    slt     $t2,$s3,$s1     #Test if $s3 and $s1 is negative?
    bltz    $s1, NEGATIVE   #Test if $s1 and $s2 is negative?
    beq     $t2,$zero,EXIT  #$s1 and $s2 are positive
    #if $s3 > $s1 then the result is not overflow
    j      OVERFLOW
NEGATIVE:
    bne     $t2,$zero,EXIT  #$s1 and $s2 aer negative
    #if $s3 < $s1 then the result is not overflow
OVERFLOW:
    li      $t0,1           #The result is overflow
EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment						Labels	
Bkpt	Address	Code	Basic	Source		Label	Address
	4194304	0x3c011001	lui \$1,4097	8: lw \$s1,x	#Load value x	as1w4.asm	
	4194308	0x8c310000	lw \$17,0(\$1)			start	4194304
	4194312	0x3c011001	lui \$1,4097	9: lw \$s2,y	#Load value y	NEGATIVE	4194360
	4194316	0x8c320004	lw \$19,4(\$1)			OVERFLOW	4194364
	4194320	0x3c011001	lui \$1,4097	10: la \$s4,isOverflow	#Load address isOverflow	EXIT	4194368
	4194324	0x34340008	ori \$20,\$1,8			x	268500992
	4194328	0x24080000	addiu \$8,\$0,0	11: li \$t0,0	#No Overflow is default status	y	268500996
	4194332	0x02329e21	addu \$19,\$17,\$18	12: addu \$s3,\$s1,\$s2	# s3 = s1 + s2	isOverflow	268501000
	4194336	0x02324826	xor \$9,\$17,\$18	13: xor \$t1,\$s1,\$s2	#Test if \$s1 and \$s2 have same sign		
	4194340	0x05200006	bltz \$9,6	15: bltz \$t1,EXIT	#If not, exit		
	4194344	0x0271502a	slt \$10,\$19,\$17	16: slt \$t2,\$s3,\$s1			
	4194348	0x04200002	bltz \$17,2	17: bltz \$s1, NEGATIVE	#Test if \$s1 and \$s2 is negative?		
	4194352	0x11400003	beq \$10,\$0,3	18: beq \$t2,\$zero,EXIT	#\$s1 and \$s2 are positive		
	4194356	0x0910000f	4194364	20: j OVERFLOW			
	4194360	0x15400001	bne \$10,\$0,1	22: bne \$t2,\$zero,EXIT	#\$s1 and \$s2 are negative		
	4194364	0x24080001	addiu \$8,\$0,1	25: li \$t0,1	#The result is overflow		
	4194368	0xaa880000	sw \$8,0(\$20)	27: sw \$t0,0(\$s4)			

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)	
268500992	2147483647	1	1	0	0	0	0	0	
268501024	0	0	0	0	0	0	0	0	
268501056	0	0	0	0	0	0	0	0	
268501088	0	0	0	0	0	0	0	0	
268501120	0	0	0	0	0	0	0	0	
268501152	0	0	0	0	0	0	0	0	
268501184	0	0	0	0	0	0	0	0	
268501216	0	0	0	0	0	0	0	0	
268501248	0	0	0	0	0	0	0	0	
268501280	0	0	0	0	0	0	0	0	
268501312	0	0	0	0	0	0	0	0	
268501344	0	0	0	0	0	0	0	0	
268501376	0	0	0	0	0	0	0	0	
268501408	0	0	0	0	0	0	0	0	

\$s1	17	2147483647
\$s2	18	1
\$s3	19	-2147483648
\$s4	20	268501000

TH5: x, y cùng âm và x+y tràn

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

.data

x: .word 0x80000000

y: .word -1

isOverflow: .word 0

.text

start:

```
lw    $s1,x           #Load value x
lw    $s2,y           #Load value y
la    $s4,isOverflow  #Load address isOverflow
li    $t0,0           #No Overflow is default status
addu  $s3,$s1,$s2     # s3 = s1 + s2
xor   $t1,$s1,$s2     #Test if $s1 and $s2 have same sign
```

```
bltz  $t1,EXIT        #If not, exit
slt   $t2,$s3,$s1
bltz  $s1, NEGATIVE   #Test if $s1 and $s2 is negative?
beq   $t2,$zero,EXIT  #If $s3 > $s1 then the result is not overflow
j     OVERFLOW
```

NEGATIVE:

```
bne   $t2,$zero,EXIT  #If $s3 < $s1 then the result is not overflow
```

OVERFLOW:

```
li    $t0,1          #The result is overflow
```

EXIT:

```
sw    $t0,0($s4)
```

- Kết quả:

Text Segment				Labels	
Offset	Address	Code	Basic	Source	Label
8	4194304	0x3c011001	lui \$1,4097	#Load value x	start
16	4194308	0x3c011000	lw \$17,0(\$1)		NEGATIVE
24	4194312	0x3c011001	lui \$1,4097		OVERFLOW
32	4194316	0x3c020004	lw \$18,4(\$1)		EXIT
40	4194320	0x3c011001	lui \$1,4097		
48	4194324	0x34340008	ori \$20,\$1,8		
56	4194328	0x24080000	addiu \$8,\$0,0		
64	4194332	0x02329821	addu \$19,\$17,\$18		
72	4194336	0x02329826	xor \$9,\$17,\$18		
80	4194340	0x05200006	bltz \$9,6		
88	4194344	0x0271502a	slt \$10,\$19,\$17		
96	4194348	0x06200002	bltz \$17,2		
104	4194352	0x11400003	beq \$10,\$0,3		
112	4194356	0x0810000f	j 4194364		
120	4194360	0x15400001	bne \$10,\$0,1		
128	4194364	0x24080001	addiu \$8,\$0,1		
136	4194368	0x0a880000	sw \$8,0(\$s4)		
144	4194372	0x00000000			

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-2147483648	-1	1	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

\$s1	17	-2147483648
\$s2	18	-1
\$s3	19	2147483647
\$s4	20	268501000

Bài 2:

a, Extract MSB of \$s0

- Chương trình:

```
#Laboratory Exercise 4, Home Assignment 2
.text
    li      $s0, 0x20194543      #load test value for these function
    andi    $t0, $s0, 0xff000000 #Extract MSB of $s0
```

- Kết quả:

Text Segment					
Bkpt	Address	Code	Basic	Source	
<input type="checkbox"/>	4194304	0x3c012019	lui \$1,0x00002019	3:	li \$s0,0x20194543 #load test value for these function
<input type="checkbox"/>	4194308	0x34304543	ori \$16,\$1,0x00004543		
<input type="checkbox"/>	4194312	0x3c01fff0	lui \$1,0xfffffffff0	4:	andi \$t0, \$s0, 0xff000000 #Extract MSB of \$s0
<input type="checkbox"/>	4194316	0x34210000	ori \$1,\$1,0x00000000		
<input type="checkbox"/>	4194320	0x02014024	and \$8,\$16,\$1		

\$t0	8	0x20000000
------	---	------------

b, Clear LSB of \$s0

- Chương trình:

```
#Laboratory Exercise 4, Home Assignment 2
.text
    li      $s0, 0x20194543      #load test value for these function
    andi    $t0, $s0, 0xfffffff0 #Clear LSB of $s0
```

- Kết quả:

Text Segment					
Bkpt	Address	Code	Basic	Source	
<input type="checkbox"/>	4194304	0x3c012019	lui \$1,0x00002019	3:	li \$s0,0x20194543 #load test value for these function
<input type="checkbox"/>	4194308	0x34304543	ori \$16,\$1,0x00004543		
<input type="checkbox"/>	4194312	0x3c01ffff	lui \$1,0xfffffffff0	4:	andi \$t0, \$s0, 0xfffffff0 #Clear LSB of \$s0
<input type="checkbox"/>	4194316	0x3421fff0	ori \$1,\$1,0x0000fff0		
<input type="checkbox"/>	4194320	0x02014024	and \$8,\$16,\$1		

\$t0	8	0x20194500
------	---	------------

c, Set LSB of \$s0(bits 7 to 0 are set to 1)

- Chương trình:

```
#Laboratory Exercise 4, Home Assignment 2
.text
    li      $s0, 0x20194543      #load test value for these function
    ori     $t0, $s0, 0x000000ff #Set LSB of $s0(bits 7 to 0 are set to 1)
```

- Kết quả:

Text Segment					
Bkpt	Address	Code	Basic	Source	
<input type="checkbox"/>	4194304	0x3c012019	lui \$1,0x00002019	3: li \$s0,0x20194543	#load test value for these function
<input type="checkbox"/>	4194308	0x34304543	ori \$16,\$1,0x00004543		
<input type="checkbox"/>	4194312	0x360800ff	ori \$8,\$16,0x000000ff	4: ori \$t0, \$s0, 0x000000ff	#Set LSB of \$s0(bits 7 to 0 are set to 1

\$t0	8	0x201945ff
------	---	------------

d, Clear \$s0 (\$s0 =0, must use logical instructions)

- Chương trình:

```
#Laboratory Exercise 4, Home Assignment 2
.text
    li      $s0, 0x20194543      #load test value for these function
    andi    $t0, $s0, 0x0        #Clear $s0($s0=0, must use logical instructions)
```

- Kết quả:

Text Segment					
Bkpt	Address	Code	Basic	Source	
<input type="checkbox"/>	4194304	0x3c012019	lui \$1,0x00002019	3: li \$s0,0x20194543	#load test value for these function
<input type="checkbox"/>	4194308	0x34304543	ori \$16,\$1,0x00004543		
<input type="checkbox"/>	4194312	0x32080000	andi \$8,\$16,0x00000000	4: andi \$t0, \$s0, 0x0	#Clear \$s0(\$s0=0, must use logical instructions)

\$t0	8	0x00000000
------	---	------------

Bài 3:

a, abs \$s0,\$s1 \$s0<= |\$s1|

- Chương trình (\$s1 <0):

```
.data
x:      .word    -10
.text
    lw      $s1, x          # load value x vao $s1
    bltz    $s1, setAbs     # if $s0<0 then jump setAbs
    add     $s0, $s1, $zero
    j       exit

setAbs:
    sub     $s0, $zero, $s1  # chuyen ve so duong (ex: 0 -- 1 =1)
    j       exit

exit:
```

- Kết quả (\$s1 <0):

Text Segment					Labels	
Bkpt	Address	Code	Basic	Source	Label	Address
	4194304	0x3c011001	lui \$1,4097	10: lw \$s1, x # load value x vao \$s1		
	4194308	0x8c310000	lw \$17,0(\$1)		as3w4.asm	
	4194312	0x06200002	bltz \$17,2	11: bltz \$s1,setAbs # if \$s0<0 then jump setAbs	setAbs	41943
	4194316	0x02200020	add \$16,\$17,\$0	12: add \$s0,\$s1,\$zero	exit	41943
	4194320	0x08100007	j 4194332	13: j exit	x	2685009
	4194324	0x0118022	sub \$16,\$0,\$17	16: sub \$s0,\$zero,\$s1 # chuyen ve so duong (ex: 0 -- 1 =1)		
	4194328	0x08100007	j 4194332	17: j exit		

Data Segment								
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	-10	0	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

\$s0	16	10
\$s1	17	-10

- Chương trình (\$s1>0):

```
.data
x:      .word    10
.text

lw      $s1, x          # load value x vao $s1
bltz    $s1,setAbs      # if $s0<0 then jump setAbs
add     $s0,$s1,$zero
j       exit

setAbs:
sub     $s0,$zero,$s1   # chuyen ve so duong (ex: 0 -- 1 =1)
j       exit

exit:
```

- Kết quả (\$s1>0):

Text Segment					Labels	
Bkpt	Address	Code	Basic	Source	Label	Address
	4194304	0x3c011001	lui \$1,4097	10: lw \$s1, x # load value x vao \$s1	as3w4.asm	
	4194308	0x8c310000	lw \$17,0(\$1)		setAbs	41943
	4194312	0x06200002	bltz \$17,2	11: bltz \$s1,setAbs # if \$s0<0 then jump setAbs	exit	41943
	4194316	0x02208020	add \$16,\$17,\$0	12: add \$s0,\$s1,\$zero	x	2685009
	4194320	0x08100007	j 4194332	13: j exit		
	4194324	0x00118022	sub \$16,\$0,\$17	16: sub \$s0,\$zero,\$s1 # chuyen ve so duong (ex: 0 -- 1 =1)		
	4194328	0x08100007	j 4194332	17: j exit		

Data Segment									
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)	
268500992	10	0	0	0	0	0	0	0	
268501024	0	0	0	0	0	0	0	0	
268501056	0	0	0	0	0	0	0	0	
268501088	0	0	0	0	0	0	0	0	
268501120	0	0	0	0	0	0	0	0	
268501152	0	0	0	0	0	0	0	0	
268501184	0	0	0	0	0	0	0	0	
268501216	0	0	0	0	0	0	0	0	
268501248	0	0	0	0	0	0	0	0	
268501280	0	0	0	0	0	0	0	0	
268501312	0	0	0	0	0	0	0	0	
268501344	0	0	0	0	0	0	0	0	
268501376	0	0	0	0	0	0	0	0	
268501408	0	0	0	0	0	0	0	0	

\$s0	16	10
\$s1	17	10

b, move \$s0,\$s1 \$s0 <=\$s1

- Chương trình:

```
.data
x:      .word    10

.text

        lw       $s1, x           # load value x vao $s1
        addu     $s0, $s1, $zero  # move $s1's value to $s0
exit:
```

- Kết quả:

\$s0	16	10
\$s1	17	10

- *Chương trình:*

```
.data
x:      .word    10

.text

        lw       $s1, x           # load value x vào $s1
        nor      $s0, $s1, $zero  # đảo bit

exit:
```

- *Kết quả:*

\$s0	16	0xffffffff5
\$s1	17	0x0000000a

+ TH2: $\$s1 = \$s2$

```

.data
x:      .word    10
y:      .word    10
.text
        lw       $s1, x           # load value x vao $s1
        lw       $s2, y           # load value y vao $s2
        slt      $t0, $s2, $s1    # if $s2 < $s1 then
        beq      $t0, $zero, label
        j        exit

label:
        j        exit
exit:

```

+ TH3: $\$s1 > \$s2$

```

.data
x:      .word    20
y:      .word    10
.text
        lw       $s1, x           # load value x vao $s1
        lw       $s2, y           # load value y vao $s2
        slt      $t0, $s2, $s1    # if $s2 < $s1 then
        beq      $t0, $zero, label
        j        exit

label:
        j        exit
exit:

```

- Kết quả:
- + TH1: $\$s1 < \$s2$

[illegible]

Segment	Address	Code	Basic	Source
	4194304	0x001003	101	1w \$a1, x # load value x wao \$a1
	4194308	0x0c310000	1w	0,(\$a1)
	4194312	0x00021000	101	2w \$a2, y # load value y wao \$a2
	4194316	0x0c310004	1w	0,(\$a1)
	4194320	0x02514020	13:	slt \$t0,\$a2,\$a1 # if \$a2 < \$a1 then
	4194324	0x11000010	14:	bne \$t0,\$zero,\$label
	4194328	0x0a100000	15c	3 exit
	4194332	0x1000005	10:	3 exit

Labels
Address

Label	Address
as3w.asm	
label	4194332
exit	4194336
x	268500992
y	268500996

☒ Data
 ☒ Text

Segment	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	10	10	0	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501040	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501160	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

Name	Number	Value
\$zero	0	0
\$at	1	268500992
\$v0	2	0
\$v1	3	0
\$a0	4	0
\$a1	5	0
\$a2	6	0
\$a3	7	0
\$t0	8	0
\$t1	9	0
\$t2	10	0
\$t3	11	0
\$t4	12	0
\$t5	13	0
\$t6	14	0
\$t7	15	0
\$t8	16	0
\$t9	17	10
\$s0	18	10
\$s1	19	0
\$s2	20	0
\$s3	21	0
\$s4	22	0
\$s5	23	0
\$s6	24	0
\$s7	25	0
\$s8	26	0
\$s9	27	0
\$fp	28	268463224
\$gp	29	2147475144
\$fp	30	0
\$ra	31	0
\$pc	0	4194320
\$t1	0	0
\$a0	0	0

Text Segment					Labels		Name		
Dispt	Address	Code	Basic	Source	Label	Address	Number	Value	
	4194304	0x0c110001	lwi \$t1, 4097	# load value x vào \$t1			0	0	
	4194308	0x0c110000	lwi \$t1, 0(\$t1)				2	269500992	
	4194312	0x0c110001	lwi \$t2, 4097	# load value y vào \$t2			3	0	
	4194316	0x0c320004	lwi \$t3, 4161		as3wLasm	4194332	4	0	
	4194320	0x0281400a	slti \$t0, \$t2, \$t1	# if \$t2 < \$t1 then	label	4194332	5	0	
	4194324	0x1100001b	beq \$t0, \$t0, \$t1		end	269500992	6	0	
	4194328	0x02810000	j \$t1		y	269500994	7	0	
	4194332	0x02810000	j \$t1				8	0	
	4194336	0x02810000	j \$t1				9	0	
	4194340	0x02810000	j \$t1				10	0	
	4194344	0x02810000	j \$t1				11	0	
	4194348	0x02810000	j \$t1				12	0	
	4194352	0x02810000	j \$t1				13	0	
	4194356	0x02810000	j \$t1				14	0	
	4194360	0x02810000	j \$t1				15	0	
	4194364	0x02810000	j \$t1				16	0	
	4194368	0x02810000	j \$t1				17	10	
	4194372	0x02810000	j \$t1				18	10	
	4194376	0x02810000	j \$t1				19	0	
	4194380	0x02810000	j \$t1				20	0	
	4194384	0x02810000	j \$t1				21	0	
	4194388	0x02810000	j \$t1				22	0	
	4194392	0x02810000	j \$t1				23	0	
	4194396	0x02810000	j \$t1				24	0	
	4194400	0x02810000	j \$t1				25	0	
	4194404	0x02810000	j \$t1				26	0	
	4194408	0x02810000	j \$t1				27	0	
	4194412	0x02810000	j \$t1				28	269468224	
	4194416	0x02810000	j \$t1				29	2147479541	
	4194420	0x02810000	j \$t1				30	0	
	4194424	0x02810000	j \$t1				31	0	
	4194428	0x02810000	j \$t1				32	4194332	
	4194432	0x02810000	j \$t1				33	0	
	4194436	0x02810000	j \$t1				34	0	
	4194440	0x02810000	j \$t1				35	0	
	4194444	0x02810000	j \$t1				36	0	
	4194448	0x02810000	j \$t1				37	0	
	4194452	0x02810000	j \$t1				38	0	
	4194456	0x02810000	j \$t1				39	0	
	4194460	0x02810000	j \$t1				40	0	
	4194464	0x02810000	j \$t1				41	0	
	4194468	0x02810000	j \$t1				42	0	
	4194472	0x02810000	j \$t1				43	0	
	4194476	0x02810000	j \$t1				44	0	
	4194480	0x02810000	j \$t1				45	0	
	4194484	0x02810000	j \$t1				46	0	
	4194488	0x02810000	j \$t1				47	0	
	4194492	0x02810000	j \$t1				48	0	
	4194496	0x02810000	j \$t1				49	0	
	4194500	0x02810000	j \$t1				50	0	
	4194504	0x02810000	j \$t1				51	0	
	4194508	0x02810000	j \$t1				52	0	
	4194512	0x02810000	j \$t1				53	0	
	4194516	0x02810000	j \$t1				54	0	
	4194520	0x02810000	j \$t1				55	0	
	4194524	0x02810000	j \$t1				56	0	
	4194528	0x02810000	j \$t1				57	0	
	4194532	0x02810000	j \$t1				58	0	
	4194536	0x02810000	j \$t1				59	0	
	4194540	0x02810000	j \$t1				60	0	
	4194544	0x02810000	j \$t1				61	0	
	4194548	0x02810000	j \$t1				62	0	
	4194552	0x02810000	j \$t1				63	0	
	4194556	0x02810000	j \$t1				64	0	
	4194560	0x02810000	j \$t1				65	0	
	4194564	0x02810000	j \$t1				66	0	
	4194568	0x02810000	j \$t1				67	0	
	4194572	0x02810000	j \$t1				68	0	
	4194576	0x02810000	j \$t1				69	0	
	4194580	0x02810000	j \$t1				70	0	
	4194584	0x02810000	j \$t1				71	0	
	4194588	0x02810000	j \$t1				72	0	
	4194592	0x02810000	j \$t1				73	0	
	4194596	0x02810000	j \$t1				74	0	
	4194600	0x02810000	j \$t1				75	0	
	4194604	0x02810000	j \$t1				76	0	
	4194608	0x02810000	j \$t1				77	0	
	4194612	0x02810000	j \$t1				78	0	
	4194616	0x02810000	j \$t1				79	0	
	4194620	0x02810000	j \$t1				80	0	
	4194624	0x02810000	j \$t1				81	0	
	4194628	0x02810000	j \$t1				82	0	
	4194632	0x02810000	j \$t1				83	0	
	4194636	0x02810000	j \$t1				84	0	
	4194640	0x02810000	j \$t1				85	0	
	4194644	0x02810000	j \$t1				86	0	
	4194648	0x02810000	j \$t1				87	0	
	4194652	0x02810000	j \$t1				88	0	
	4194656	0x02810000	j \$t1				89	0	
	4194660	0x02810000	j \$t1				90	0	
	4194664	0x02810000	j \$t1				91	0	
	4194668	0x02810000	j \$t1				92	0	
	4194672	0x02810000	j \$t1				93	0	
	4194676	0x02810000	j \$t1				94	0	
	4194680	0x02810000	j \$t1				95	0	
	4194684	0x02810000	j \$t1				96	0	
	4194688	0x02810000	j \$t1				97	0	
	4194692	0x02810000	j \$t1				98	0	
	4194696	0x02810000	j \$t1				99	0	
	4194700	0x02810000	j \$t1				100	0	
	4194704	0x02810000	j \$t1				101	0	
	4194708	0x02810000	j \$t1				102	0	
	4194712	0x02810000	j \$t1				103	0	
	4194716	0x02810000	j \$t1				104	0	
	4194720	0x02810000	j \$t1				105	0	
	4194724	0x02810000	j \$t1				106	0	
	4194728	0x02810000	j \$t1				107	0	
	4194732	0x02810000	j \$t1				108	0	
	4194736	0x02810000	j \$t1				109	0	
	4194740	0x02810000	j \$t1				110	0	
	4194744	0x02810000	j \$t1				111	0	
	4194748	0x02810000	j \$t1				112	0	
	4194752	0x02810000	j \$t1				113	0	
	4194756	0x02810000	j \$t1				114	0	
	4194760	0x02810000	j \$t1				115	0	
	4194764	0x02810000	j \$t1				116	0	
	4194768	0x02810000	j \$t1				117	0	
	4194772	0x02810000	j \$t1				118	0	
	4194776	0x02810000	j \$t1				119	0	
	4194780	0x02810000	j \$t1				120	0	
	4194784	0x02810000	j \$t1				121	0	
	4194788	0x02810000	j \$t1				122	0	
	4194792	0x02810000	j \$t1				123	0	
	4194796	0x02810000	j \$t1				124	0	
	4194800	0x02810000	j \$t1				125	0	
	4194804	0x02810000	j \$t1				126	0	
	4194808	0x02810000	j \$t1				127	0	
	4194812	0x02810000	j \$t1				128	0	
	4194816	0x02810000	j \$t1				129	0	
	4194820	0x02810000	j \$t1				130	0	
	4194824	0x02810000	j \$t1				131	0	
	4194828	0x02810000	j \$t1				132	0	
	4194832	0x02810000	j \$t1				133	0	
	4194836	0x02810000	j \$t1				134	0	
	4194840	0x02810000	j \$t1				135	0	
	4194844	0x02810000	j \$t1				136	0	
	4194848	0x02810000	j \$t1				137	0	
	4194852	0x02810000	j \$t1				138	0	
	4194856	0x02810000	j \$t1				139	0	
	4194860	0x02810000	j \$t1				140	0	
	4194864	0x02810000	j \$t1				141	0	
	4194868	0x02810000	j \$t1				142	0	
	4194872	0x02810000	j \$t1				143	0	
	4194876	0x02810000	j \$t1				144	0	
	4194880	0x02810000	j \$t1				145	0	
	4194884	0x02810000	j \$t1				146	0	
	4194888	0x02810000	j \$t1				147	0	
	4194892	0x02810000	j \$t1				148	0	
	4194896	0x02810000	j \$t1				149	0	
	4194900	0x02810000	j \$t1				150	0	
	4194904	0x02810000	j \$t1						

```

#Laboratory Exercise 4, Home Assignment 1
.data
x:          .word   -235375
y:          .word   357925
isOverflow: .word    0
.text
start:
    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2      # s3 = s1 + s2
    xor     $t1,$s1,$s2      #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT         #If not, exit
    xor     $t2,$s2,$s3      #Test if $s2 and $s3 have same sign
    bltz    $t2,OVERFLOW    #If not ,jump overflow
    j       EXIT

OVERFLOW:
    li      $t0,1           #The result is overflow
EXIT:
    sw      $t0,0($s4)

```

TH1: x,y trái dấu

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:                .word    -235375
y:                .word    357925
isOverflow:       .word    0

.text
start:

    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2     # s3 = s1 + s2
    xor     $t1,$s1,$s2     #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT        #If not, exit
    xor     $t2,$s2,$s3     #Test if $s2 and $s3 have same sign
    bltz    $t2,OVERFLOW   #If not ,jump overflow
    j       EXIT

OVERFLOW:
    li      $t0,1          #The result is overflow

EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment

Byte

Address

Code

Basic

Source

Flow value x

4194304

0x3011001

lui

\$1,4097

4194308

0x3010000

lw

\$17,0(\$1)

4194312

0x3011001

lui

\$16,4097

4194316

0x3012000

lw

\$16,0(\$1)

4194320

0x3011001

lui

\$1,4097

4194324

0x1400000

addiu

\$2,\$2,0

4194328

0x2400000

addiu

\$2,\$2,0

4194332

0x0202921

addu

\$3,\$2,\$2

\$3 = \$2 + \$2

4194336

0x0202921

addu

\$3,\$2,\$2

\$3not if \$2 and \$2 have same sign

4194340

0x0202921

bits

\$17,\$17

\$17 not , exit

4194344

0x0202921

addu

\$3,\$2,\$2

\$3not if \$2 and \$2 have same sign

4194348

0x0202921

bits

\$17,\$17

\$17 not , cmp overflow

4194352

0x0200000

bits

\$17,\$17

\$17 not , cmp overflow

4194356

0x2400000

addiu

\$2,\$2,0

4194360

0x0200000

addiu

\$2,\$2,0

\$2The result is overflow

Labels

Label

Address

start

4194304

OVERFLOW

4194316

EXIT

4194340

Y

265500992

isOverflow

265500996

265501000

Name

Number

Value

\$2ero

0

0

\$25

2

265500992

\$17

3

0

\$16

4

0

\$1

5

0

\$3

6

0

\$2

7

0

\$17

8

-450592

\$2

9

0

\$3

10

0

\$17

11

0

\$2

12

0

\$3

13

0

\$17

14

0

\$2

15

0

\$1

16

0

\$2

17

-235375

\$3

18

357925

\$2

19

123151

\$4

20

265500992

\$17

21

0

\$2

22

0

\$17

23

0

\$2

24

0

\$17

25

0

\$2

26

0

\$17

27

0

\$2

28

0

\$17

29

2147479540

\$2

30

0

\$17

31

0

\$2

32

0

\$17

33

0

\$2

34

0

\$17

35

0

\$2

36

0

\$17

37

0

\$2

38

0

\$17

39

0

\$2

40

0

\$17

41

0

\$2

42

0

\$17

43

0

\$2

44

0

\$17

45

0

\$2

46

0

\$17

47

0

\$2

48

0

\$17

49

0

\$2

50

0

\$17

51

0

\$2

52

0

\$17

53

0

\$2

54

0

\$17

55

0

\$2

56

0

\$17

57

0

\$2

58

0

\$17

59

0

\$2

60

0

\$17

61

0

\$2

62

0

\$17

63

0

\$2

64

0

\$17

65

0

\$2

66

0

\$17

67

0

\$2

68

0

\$17

69

0

\$2

70

0

\$17

71

0

\$2

72

0

\$17

73

0

\$2

74

0

\$17

75

0

\$2

76

0

\$17

77

0

\$2

78

0

\$17

79

0

\$2

80

0

\$17

81

0

\$2

82

0

\$17

83

0

\$2

84

0

\$17

85

0

\$2

86

0

\$17

87

0

\$2

88

0

\$17

89

0

\$2

90

0

\$17

91

0

\$2

92

0

\$17

93

0

\$2

94

0

\$17

95

0

\$2

96

0

\$17

97

0

\$2

98

0

\$17

99

0

\$2

100

0

Data Segment

Address

Value (-)

Value (+)

Value (+)

Value (+)

Value (+)

Value (+)

Value (+)

Value (+)

265500992

-235375

357925

0

0

0

0

0

0

265501024

0

0

0

0

0

0

0

0

265501040

0

0

0

0

0

0

0

0

265501088

0

0

0

0

0

0

0

0

265501120

0

0

0

0

0

0

0

0

265501152

0

0

0

0

0

0

0

0

265501184

0

0

0

0

0

0

0

0

265501216

0

0

0

0

0

0

0

0

265501248

0

0

0

0

0

0

0

0

265501280

0

0

0

0

0

0

0

0

265501312

0

0

0

0

0

0

0

0

265501344

0

0

0

0

0

0

0

0

265501376

0

0

0

0

0

0

0

0

265501408

0

0

0

0

0

0

0

0

✓ Data

✓ Text

TH2: x,y cùng dương, không tràn

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:                .word    235375
y:                .word    357925
isOverflow:       .word    0
.text
start:
    lw            $s1,x                #Load value x
    lw            $s2,y                #Load value y
    la            $s4,isOverflow       #Load address isOverflow
    li            $t0,0                #No Overflow is default status
    addu          $s3,$s1,$s2          # s3 = s1 + s2
    xor           $t1,$s1,$s2          #Test if $s1 and $s2 have same sign

    bltz          $t1,EXIT             #If not, exit
    xor           $t2,$s2,$s3          #Test if $s2 and $s3 have same sign
    bltz          $t2,OVERFLOW         #If not ,jump overflow
    j             EXIT

OVERFLOW:
    li            $t0,1                #The result is overflow

EXIT:
    sw            $t0,0($s4)
```

- Kết quả:

Text Segment	Address	Code	Basic	Source	Labels	Name	Number	Value
4194374	0x3e11001	lw	\$s1,x	#Load value x	start	4194304	0	0
4194380	0x3e11000	lw	\$s2,y	#Load value y	OVERFLOW	4194354	1	26550992
4194312	0x3e11000	la	\$s4,isOverflow	#Load address isOverflow	EXIT	4194360	2	0
4194320	0x3e11001	li	\$t0,0	#No Overflow is default status	x	26550992	3	0
4194324	0x3e11000	addu	\$s3,\$s1,\$s2	# s3 = s1 + s2	isOverflow	26550994	4	0
4194332	0x2121921	xor	\$t1,\$s1,\$s2	#Test if \$s1 and \$s2 have same sign			5	0
4194336	0x2121921	bltz	\$t1,EXIT	#If not, exit			6	0
4194340	0x2121921	xor	\$t2,\$s2,\$s3	#Test if \$s2 and \$s3 have same sign			7	0
4194344	0x2121921	bltz	\$t2,OVERFLOW	#If not ,jump overflow			8	0
4194348	0x2121921	j	EXIT				9	450890
4194352	0x2121921	li	\$t0,1	#The result is overflow			10	82899
4194356	0x2121921	sw	\$t0,0(\$s4)				11	0
4194360	0x2121921						12	0
4194364	0x2121921						13	0
4194368	0x2121921						14	0
4194372	0x2121921						15	0
4194376	0x2121921						16	0
4194380	0x2121921						17	235375
4194384	0x2121921						18	357925
4194388	0x2121921						19	357925
4194392	0x2121921						20	26550992
4194396	0x2121921						21	0
4194400	0x2121921						22	0
4194404	0x2121921						23	0
4194408	0x2121921						24	0
4194412	0x2121921						25	0
4194416	0x2121921						26	0
4194420	0x2121921						27	0
4194424	0x2121921						28	26550992
4194428	0x2121921						29	2147479541
4194432	0x2121921						30	0
4194436	0x2121921						31	0
4194440	0x2121921						32	4194364
4194444	0x2121921						33	0
4194448	0x2121921						34	0
4194452	0x2121921						35	0
4194456	0x2121921						36	0
4194460	0x2121921						37	0
4194464	0x2121921						38	0
4194468	0x2121921						39	0
4194472	0x2121921						40	0
4194476	0x2121921						41	0
4194480	0x2121921						42	0
4194484	0x2121921						43	0
4194488	0x2121921						44	0
4194492	0x2121921						45	0
4194496	0x2121921						46	0
4194500	0x2121921						47	0
4194504	0x2121921						48	0
4194508	0x2121921						49	0
4194512	0x2121921						50	0
4194516	0x2121921						51	0
4194520	0x2121921						52	0
4194524	0x2121921						53	0
4194528	0x2121921						54	0
4194532	0x2121921						55	0
4194536	0x2121921						56	0
4194540	0x2121921						57	0
4194544	0x2121921						58	0
4194548	0x2121921						59	0
4194552	0x2121921						60	0
4194556	0x2121921						61	0
4194560	0x2121921						62	0
4194564	0x2121921						63	0
4194568	0x2121921						64	0
4194572	0x2121921						65	0
4194576	0x2121921						66	0
4194580	0x2121921						67	0
4194584	0x2121921						68	0
4194588	0x2121921						69	0
4194592	0x2121921						70	0
4194596	0x2121921						71	0
4194600	0x2121921						72	0
4194604	0x2121921						73	0
4194608	0x2121921						74	0
4194612	0x2121921						75	0
4194616	0x2121921						76	0
4194620	0x2121921						77	0
4194624	0x2121921						78	0
4194628	0x2121921						79	0
4194632	0x2121921						80	0
4194636	0x2121921						81	0
4194640	0x2121921						82	0
4194644	0x2121921						83	0
4194648	0x2121921						84	0
4194652	0x2121921						85	0
4194656	0x2121921						86	0
4194660	0x2121921						87	0
4194664	0x2121921						88	0
4194668	0x2121921						89	0
4194672	0x2121921						90	0
4194676	0x2121921						91	0
4194680	0x2121921						92	0
4194684	0x2121921						93	0
4194688	0x2121921						94	0
4194692	0x2121921						95	0
4194696	0x2121921						96	0
4194700	0x2121921						97	0
4194704	0x2121921						98	0
4194708	0x2121921						99	0
4194712	0x2121921						100	0

TH3: x,y cùng âm, không tràn

- Chương trình:

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:          .word    -235375
y:          .word    -357925
isOverflow: .word    0

.text
start:

    lw      $s1,x          #Load value x
    lw      $s2,y          #Load value y
    la      $s4,isOverflow #Load address isOverflow
    li      $t0,0          #No Overflow is default status
    addu    $s3,$s1,$s2    # s3 = s1 + s2
    xor     $t1,$s1,$s2    #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT       #If not, exit
    xor     $t2,$s2,$s3    #Test if $s2 and $s3 have same sign
    bltz    $t2,OVERFLOW  #If not ,jump overflow
    j       EXIT

OVERFLOW:
    li      $t0,1          #The result is overflow

EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment				Labels		Name			Number	Value
Expr	Address	Code	Basic	Source	Label	Address				
4194304	0x3011000	lw	\$t1,x	#Load value x	start	4194304	\$zero	0	0	
4194308	0x3011000	lw	\$t2,y	#Load value y	OVERFLOW	4194304	\$t0	1	26850092	
4194312	0x3011000	la	\$s4,isOverflow	#Load address isOverflow	EXIT	4194304	\$t1	2	0	
4194320	0x3011000	li	\$t0,0	#No Overflow is default status	EXIT	26850092	\$t2	3	0	
4194324	0x2400000	addu	\$s3,\$s1,\$s2	# s3 = s1 + s2	isOverflow	26850092	\$s0	4	0	
4194328	0x2400000	xor	\$t1,\$s1,\$s2	#Test if \$s1 and \$s2 have same sign		26850092	\$s1	5	0	
4194332	0x2129221	bltz	\$t1,EXIT	#If not, exit		26850092	\$s2	6	0	
4194336	0x2024000	xor	\$t2,\$s2,\$s3	#Test if \$s2 and \$s3 have same sign		26850092	\$s3	7	0	
4194340	0x7200004	bltz	\$t2,OVERFLOW	#If not ,jump overflow		26850092	\$t0	8	0	
4194344	0x0000000	j	EXIT			26850092	\$t1	9	450890	
4194352	0x0020000	sw	\$t0,0(\$s4)	#The result is overflow		26850092	\$t2	10	33933	
4194356	0x2400000	sw	\$s3,0(\$s4)			26850092	\$t3	11	0	
4194360	0x0000000	sw	\$t0,0(\$s4)			26850092	\$t4	12	0	
Data Segment										
Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)		
26850092	-235375	-357925	0	0	0	0	0	0	0	0
26850104	0	0	0	0	0	0	0	0	0	0
26850116	0	0	0	0	0	0	0	0	0	0
26850108	0	0	0	0	0	0	0	0	0	0
26850120	0	0	0	0	0	0	0	0	0	0
26850132	0	0	0	0	0	0	0	0	0	0
26850114	0	0	0	0	0	0	0	0	0	0
26850126	0	0	0	0	0	0	0	0	0	0
26850140	0	0	0	0	0	0	0	0	0	0
26850128	0	0	0	0	0	0	0	0	0	0
26850132	0	0	0	0	0	0	0	0	0	0
26850144	0	0	0	0	0	0	0	0	0	0
26850136	0	0	0	0	0	0	0	0	0	0
26850148	0	0	0	0	0	0	0	0	0	0
26850140	0	0	0	0	0	0	0	0	0	0

#Laboratory Exercise 4, Home Assignment 1

```
.data
x:          .word    0x7fffffff
y:          .word    1
isOverflow: .word    0

.text
start:

    lw      $s1,x          #Load value x
    lw      $s2,y          #Load value y
    la      $s4,isOverflow #Load address isOverflow
    li      $t0,0          #No Overflow is default status
    addu    $s3,$s1,$s2    # s3 = s1 + s2
    xor     $t1,$s1,$s2    #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT       #If not, exit
    xor     $t2,$s2,$s3    #Test if $s2 and $s3 have same sign
    bltz    $t2,OVERFLOW  #If not ,jump overflow
    j       EXIT

OVERFLOW:
    li      $t0,1          #The result is overflow

EXIT:
    sw      $t0,0($s4)
```

- Kết quả:

Text Segment

Bkpt	Address	Code	Basic	Source
	4194304	0x3c011001	lui \$1,4097	8: lw \$s1,x #Load value x
	4194308	0x8c310000	lw \$17,0(\$1)	
	4194312	0x3c011001	lui \$1,4097	9: lw \$s2,y #Load value y
	4194316	0x8c320004	lw \$18,4(\$1)	
	4194320	0x3c011001	lui \$1,4097	10: la \$s4,isOverflow #Load address isOverflow
	4194324	0x34340008	ori \$20,\$1,8	
	4194328	0x24080000	addiu \$8,\$0,0	11: li \$t0,0 #No Overflow is default status
	4194332	0x02329821	addu \$19,\$17,\$18	12: addu \$s3,\$s1,\$s2 # s3 = s1 + s2
	4194336	0x02324826	xor \$9,\$17,\$18	13: xor \$t1,\$s1,\$s2 #Test if \$s1 and \$s2 have same sign
	4194340	0x05200004	bltz \$9,4	15: bltz \$t1,EXIT #if not, exit
	4194344	0x02535026	xor \$10,\$18,\$19	16: xor \$t2,\$s2,\$s3 #Test if \$s2 and \$s3 have same sign
	4194348	0x05400001	bltz \$10,1	17: bltz \$t2,OVERFLOW #if not ,jump overflow
	4194352	0x0810000e	j 4194360	18: j EXIT
	4194356	0x24080001	addiu \$8,\$0,1	20: li \$t0,1 #The result is overflow
	4194360	0xae880000	sw \$8,0(\$20)	22: sw \$t0,0(\$s4)

Labels

Label	Address
as1w4.asm	4194304
start	4194356
OVERFLOW	4194360
EXIT	268500992
x	268500996
y	268501000
isOverflow	268501000

☒ Data ☒ Text

Data Segment

Address	Value (+0)	Value (+4)	Value (+8)	Value (+12)	Value (+16)	Value (+20)	Value (+24)	Value (+28)
268500992	2147483647	1	1	0	0	0	0	0
268501024	0	0	0	0	0	0	0	0
268501056	0	0	0	0	0	0	0	0
268501088	0	0	0	0	0	0	0	0
268501120	0	0	0	0	0	0	0	0
268501152	0	0	0	0	0	0	0	0
268501184	0	0	0	0	0	0	0	0
268501216	0	0	0	0	0	0	0	0
268501248	0	0	0	0	0	0	0	0
268501280	0	0	0	0	0	0	0	0
268501312	0	0	0	0	0	0	0	0
268501344	0	0	0	0	0	0	0	0
268501376	0	0	0	0	0	0	0	0
268501408	0	0	0	0	0	0	0	0

TH5: x,y cùng âm, tràn

- Chương trình:

```
.data
x:          .word    0x80000000
y:          .word    -1
isOverflow: .word    0

.text
start:

    lw      $s1,x           #Load value x
    lw      $s2,y           #Load value y
    la      $s4,isOverflow  #Load address isOverflow
    li      $t0,0           #No Overflow is default status
    addu    $s3,$s1,$s2     # s3 = s1 + s2
    xor     $t1,$s1,$s2     #Test if $s1 and $s2 have same sign

    bltz    $t1,EXIT        #If not, exit
    xor     $t2,$s2,$s3     #Test if $s2 and $s3 have same sign
    bltz    $t2,OVERFLOW    #If not ,jump overflow
    j       EXIT

OVERFLOW:
    li      $t0,1           #The result is overflow

EXIT:
    sw      $t0,0($s4)
```

- *Kết quả:*

Text Segment				Labels		Name			Number			Value		
Expr	Address	Code	Basic	Source	Label	Address								
	6194101	0x00100010	int 51,0x0									0	0	
	6194108	0x00100000	lw 617,0x0	Fixed value x	asfwwasm							269500992		
	6194112	0x00001000	int 51,0x0									0	0	
	6194116	0x00200000	lw 617,0x0	Fixed value y		41943004						0	0	
	6194120	0x00200000	lw 617,0x0	Fixed value z		41943004						0	0	
	6194124	0x00001000	int 51,0x0									0	0	
	6194128	0x00340000	ori 620,0x0									0	0	
	6194132	0x00340000	addui 619,617,0x0									0	0	
	6194136	0x003239e2	addu 619,617,0x0									0	0	
	6194140	0x003239e2	nor 619,617,0x0									0	0	
	6194144	0x00200000	addu 619,617,0x0									0	0	
	6194148	0x003239e2	nor 619,617,0x0									0	0	
	6194152	0x00340000	bltu 619,617,0x0									0	0	
	6194156	0x00340000	addu 619,617,0x0									0	0	
	6194160	0x00340000	addu 619,617,0x0									0	0	
	6194164	0x00340000	addu 619,617,0x0									0	0	
	6194168	0x00340000	addu 619,617,0x0									0	0	
	6194172	0x00340000	addu 619,617,0x0									0	0	
	6194176	0x00340000	addu 619,617,0x0									0	0	
	6194180	0x00340000	addu 619,617,0x0									0	0	
	6194184	0x00340000	addu 619,617,0x0									0	0	
	6194188	0x00340000	addu 619,617,0x0									0	0	
	6194192	0x00340000	addu 619,617,0x0									0	0	
	6194196	0x00340000	addu 619,617,0x0									0	0	
	6194200	0x00340000	addu 619,617,0x0									0	0	
	6194204	0x00340000	addu 619,617,0x0									0	0	
	6194208	0x00340000	addu 619,617,0x0									0	0	
	6194212	0x00340000	addu 619,617,0x0									0	0	
	6194216	0x00340000	addu 619,617,0x0									0	0	
	6194220	0x00340000	addu 619,617,0x0									0	0	
	6194224	0x00340000	addu 619,617,0x0									0	0	
	6194228	0x00340000	addu 619,617,0x0									0	0	
	6194232	0x00340000	addu 619,617,0x0									0	0	
	6194236	0x00340000	addu 619,617,0x0									0	0	
	6194240	0x00340000	addu 619,617,0x0									0	0	
	6194244	0x00340000	addu 619,617,0x0									0	0	
	6194248	0x00340000	addu 619,617,0x0									0	0	
	6194252	0x00340000	addu 619,617,0x0									0	0	
	6194256	0x00340000	addu 619,617,0x0									0	0	
	6194260	0x00340000	addu 619,617,0x0									0	0	
	6194264	0x00340000	addu 619,617,0x0									0	0	
	6194268	0x00340000	addu 619,617,0x0									0	0	
	6194272	0x00340000	addu 619,617,0x0									0	0	
	6194276	0x00340000	addu 619,617,0x0									0	0	
	6194280	0x00340000	addu 619,617,0x0									0	0	
	6194284	0x00340000	addu 619,617,0x0									0	0	
	6194288	0x00340000	addu 619,617,0x0									0	0	
	6194292	0x00340000	addu 619,617,0x0									0	0	
	6194296	0x00340000	addu 619,617,0x0									0	0	
	6194300	0x00340000	addu 619,617,0x0									0	0	
	6194304	0x00340000	addu 619,617,0x0									0	0	
	6194308	0x00340000	addu 619,617,0x0									0	0	
	6194312	0x00340000	addu 619,617,0x0									0	0	
	6194316	0x00340000	addu 619,617,0x0									0	0	
	6194320	0x00340000	addu 619,617,0x0									0	0	
	6194324	0x00340000	addu 619,617,0x0									0	0	
	6194328	0x00340000	addu 619,617,0x0									0	0	
	6194332	0x00340000	addu 619,617,0x0									0	0	
	6194336	0x00340000	addu 619,617,0x0									0	0	
	6194340	0x00340000	addu 619,617,0x0									0	0	
	6194344	0x00340000	addu 619,617,0x0									0	0	
	6194348	0x00340000	addu 619,617,0x0									0	0	
	6194352	0x00340000	addu 619,617,0x0									0	0	
	6194356	0x00340000	addu 619,617,0x0									0	0	
	6194360	0x00340000	addu 619,617,0x0									0	0	
	6194364	0x00340000	addu 619,617,0x0									0	0	
	6194368	0x00340000	addu 619,617,0x0									0	0	
	6194372	0x00340000	addu 619,617,0x0									0	0	
	6194376	0x00340000	addu 619,617,0x0									0	0	
	6194380	0x00340000	addu 619,617,0x0									0	0	
	6194384	0x00340000	addu 619,617,0x0									0	0	
	6194388	0x00340000	addu 619,617,0x0									0	0	
	6194392	0x00340000	addu 619,617,0x0									0	0	
	6194396	0x00340000	addu 619,617,0x0									0	0	
	6194400	0x00340000	addu 619,617,0x0									0	0	
	6194404	0x00340000	addu 619,617,0x0									0	0	
	6194408	0x00340000	addu 619,617,0x0									0	0	
	6194412	0x00340000	addu 619,617,0x0									0	0	
	6194416	0x00340000	addu 619,617,0x0									0	0	
	6194420	0x00340000	addu 619,617,0x0									0	0	
	6194424	0x00340000	addu 619,617,0x0									0	0	
	6194428	0x00340000	addu 619,617,0x0									0	0	
	6194432	0x00340000	addu 619,617,0x0									0	0	
	6194436	0x00340000	addu 619,617,0x0									0	0	
	6194440	0x00340000	addu 619,617,0x0									0	0	
	6194444	0x00340000	addu 619,617,0x0									0	0	
	6194448	0x00340000	addu 619,617,0x0									0	0	
	6194452	0x00340000	addu 619,617,0x0									0	0	
	6194456	0x00340000	addu 619,617,0x0									0	0	
	6194460	0x00340000	addu 619,617,0x0									0	0	
	6194464	0x00340000	addu 619,617,0x0									0	0	
	6194468	0x00340000	addu 619,617,0x0									0	0	
	6194472	0x00340000	addu 619,617,0x0									0	0	
	6194476	0x00340000	addu 619,617,0x0									0	0	
	6194480	0x00340000	addu 619,617,0x0									0	0	
	6194484	0x00340000	addu 619,617,0x0									0	0	
	6194488	0x00340000	addu 619,617,0x0									0	0	
	6194492	0x00340000	addu 619,617,0x0									0	0	
	6194496	0x00340000	addu 619,617,0x0									0	0	
	6194500	0x00340000	addu 619,617,0x0									0	0	
	6194504	0x00340000	addu 619,617,0x0									0	0	
	6194508	0x00340000	addu 619,617,0x0									0	0	
	6194512	0x00340000	addu 619,617,0x0									0	0	
	6194516	0x00340000	addu 619,617,0x0									0	0	
	6194520	0x00340000	addu 619,617,0x0									0	0	
	6194524	0x00340000	addu 619,617,0x0									0	0	
	6194528	0x00340000	addu 619,617,0x0									0	0	
	6194532	0x00340000	addu 619,617,0x0									0	0	
	6194536	0x00340000	addu 619,617,0x0									0	0	
	6194540	0x00340000	addu 619,617,0x0									0	0	
	6194544	0x00340000	addu 619,617,0x0									0	0	
	6194548	0x00340000	addu 619,617,0x0									0	0	
	6194552	0x00340000	addu 619,617,0x0									0	0	
	6194556	0x00340000	addu 619,617,0x0									0	0	
	6194560	0x00340000	addu 619,617,0x0									0	0	
	6194564	0x00340000	addu 619,617,0x0									0	0	
	6194568	0x00340000	addu 619,617,0x0									0	0	
	6194572	0x00340000	addu 619,617,0x0	</										

Bài 5:

- *Chương trình:*

#Laboratory Exercise 5

.data

```
x:      .word    12      #x=12
y:      .word    4      #y=4
i:      .word    0      # bien chay i=0
sum:    .word    0      # tich sum=0
```

.text

```
lw      $s0,x      # load value x
lw      $s1,y      # load value y
lw      $s2,i      # load value i
la      $s3,sum     # load address sum
li      $s4,1      # load 1 to $s4

loop:
beq     $s1,$s4,endoop # if y=1 then endloop
srl     $s1,$s1,1      # y=y:2
addi    $s2,$s2,1      # i=i+1
j       loop

endoop:
sllv    $s5,$s0,$s2    # sum=x*y
sw      $s5,0($s3)     # return sum
```

- Kết quả:

Text Segment					Labels		Registers		
Bit	Address	Code	Basic	Source	Label	Address	Name	Number	Value
	4194304	0x30110001	lw \$t0,x	# load value x			\$zero	0	0
	4194308	0x30100001	lw \$t1,y	# load value y			\$t0	1	269500992
	4194312	0x30110001	lw \$t2,i	# load value i			\$t1	2	0
	4194316	0x30110004	la \$t3,sum	# load address sum			\$t2	3	0
	4194320	0x30110001	li \$t4,1	# load 1 to \$t4			\$t3	4	0
	4194324	0x30100001	beq \$t1,\$t4,endoop	# if y=1 then endloop	loop	4194340	\$t4	5	0
	4194328	0x30100001	srl \$t1,\$t1,1	# y=y:2	endoop	4194356	\$t5	6	0
	4194332	0x30100001	addi \$t2,\$t2,1	# i=i+1			\$t6	7	0
	4194336	0x30110001	j loop				\$t7	8	0
	4194340	0x30100001					\$t8	9	0
	4194344	0x30100001					\$t9	10	0
	4194348	0x30100001					\$t10	11	0
	4194352	0x30100001					\$t11	12	0
	4194356	0x30100001					\$t12	13	0
	4194360	0x30100001					\$t13	14	0
	4194364	0x30100001					\$t14	15	0
	4194368	0x30100001					\$t15	16	0
	4194372	0x30100001					\$t16	17	0
	4194376	0x30100001					\$t17	18	0
	4194380	0x30100001					\$t18	19	0
	4194384	0x30100001					\$t19	20	0
	4194388	0x30100001					\$t20	21	0
	4194392	0x30100001					\$t21	22	0
	4194396	0x30100001					\$t22	23	0
	4194400	0x30100001					\$t23	24	0
	4194404	0x30100001					\$t24	25	0
	4194408	0x30100001					\$t25	26	0
	4194412	0x30100001					\$t26	27	0
	4194416	0x30100001					\$t27	28	0
	4194420	0x30100001					\$t28	29	0
	4194424	0x30100001					\$t29	30	0
	4194428	0x30100001					\$t30	31	0
	4194432	0x30100001					\$t31	32	0
	4194436	0x30100001					\$t32	33	0
	4194440	0x30100001					\$t33	34	0
	4194444	0x30100001					\$t34	35	0
	4194448	0x30100001					\$t35	36	0
	4194452	0x30100001					\$t36	37	0
	4194456	0x30100001					\$t37	38	0
	4194460	0x30100001					\$t38	39	0
	4194464	0x30100001					\$t39	40	0
	4194468	0x30100001					\$t40	41	0
	4194472	0x30100001					\$t41	42	0
	4194476	0x30100001					\$t42	43	0
	4194480	0x30100001					\$t43	44	0
	4194484	0x30100001					\$t44	45	0
	4194488	0x30100001					\$t45	46	0
	4194492	0x30100001					\$t46	47	0
	4194496	0x30100001					\$t47	48	0
	4194500	0x30100001					\$t48	49	0
	4194504	0x30100001					\$t49	50	0
	4194508	0x30100001					\$t50	51	0
	4194512	0x30100001					\$t51	52	0
	4194516	0x30100001					\$t52	53	0
	4194520	0x30100001					\$t53	54	0
	4194524	0x30100001					\$t54	55	0
	4194528	0x30100001					\$t55	56	0
	4194532	0x30100001					\$t56	57	0
	4194536	0x30100001					\$t57	58	0
	4194540	0x30100001					\$t58	59	0
	4194544	0x30100001					\$t59	60	0
	4194548	0x30100001					\$t60	61	0
	4194552	0x30100001					\$t61	62	0
	4194556	0x30100001					\$t62	63	0
	4194560	0x30100001					\$t63	64	0
	4194564	0x30100001					\$t64	65	0
	4194568	0x30100001					\$t65	66	0
	4194572	0x30100001					\$t66	67	0
	4194576	0x30100001					\$t67	68	0
	4194580	0x30100001					\$t68	69	0
	4194584	0x30100001					\$t69	70	0
	4194588	0x30100001					\$t70	71	0
	4194592	0x30100001					\$t71	72	0
	4194596	0x30100001					\$t72	73	0
	4194600	0x30100001					\$t73	74	0
	4194604	0x30100001					\$t74	75	0
	4194608	0x30100001					\$t75	76	0
	4194612	0x30100001					\$t76	77	0
	4194616	0x30100001					\$t77	78	0
	4194620	0x30100001					\$t78	79	0
	4194624	0x30100001					\$t79	80	0
	4194628	0x30100001					\$t80	81	0
	4194632	0x30100001					\$t81	82	0
	4194636	0x30100001					\$t82	83	0
	4194640	0x30100001					\$t83	84	0
	4194644	0x30100001					\$t84	85	0
	4194648	0x30100001					\$t85	86	0
	4194652	0x30100001					\$t86	87	0
	4194656	0x30100001					\$t87	88	0
	4194660	0x30100001					\$t88	89	0
	4194664	0x30100001					\$t89	90	0
	4194668	0x30100001					\$t90	91	0
	4194672	0x30100001					\$t91	92	0
	4194676	0x30100001					\$t92	93	0
	4194680	0x30100001					\$t93	94	0
	4194684	0x30100001					\$t94	95	0
	4194688	0x30100001					\$t95	96	0
	4194692	0x30100001					\$t96	97	0
	4194696	0x30100001					\$t97	98	0
	4194700	0x30100001					\$t98	99	0
	4194704	0x30100001					\$t99	100	0
	4194708	0x30100001					\$t100	101	0
	4194712	0x30100001					\$t101	102	0
	4194716	0x30100001					\$t102	103	0
	4194720	0x30100001					\$t103	104	0
	4194724	0x30100001					\$t104	105	0
	4194728	0x30100001					\$t105	106	0
	4194732	0x30100001					\$t106	107	0
	4194736	0x30100001					\$t107	108	0
	4194740	0x30100001					\$t108	109	0
	4194744	0x30100001					\$t109	110	0
	4194748	0x30100001					\$t110	111	0
	4194752	0x30100001					\$t111	112	0
	4194756	0x30100001					\$t112	113	0
	4194760	0x30100001					\$t113	114	0
	4194764	0x30100001					\$t114	115	0
	4194768	0x30100001					\$t115	116	0
	4194772	0x30100001					\$t116	117	0
	4194776	0x30100001					\$t117	118	0
	4194780	0x30100001					\$t118	119	0
	4194784	0x30100001					\$t119	120	0
	4194788	0x30100001					\$t120	121	0
	4194792	0x30100001					\$t121	122	0
	4194796	0x30100001					\$t122	123	0
	4194800	0x30100001					\$t123	124	0
	4194804	0x30100001					\$t124	125	0
	4194808	0x30100001					\$t125	126	0
	4194812	0x30100001					\$t126	127	0
	4194816	0x30100001					\$t127	128	0
	4194820	0x30100001					\$t128	129	0
	4194824	0x30100001					\$t129	130	0
	4194828	0x30100001					\$t130	131	0
	4194832	0x30100001					\$t131	132	0
	4194836	0x30100001					\$t132	133	0
	4194840	0x30100001					\$t133	134	0
	4194844	0x30100001					\$t134	135	0
	4194848	0x30100001					\$t135	136	0
	4194852	0x30100001					\$t136	137	0
	4194856	0x30100001					\$t137	138	0
	4194860	0x30100001					\$t138	139	0
	4194864	0x30100001					\$t139	140	0
	4194868	0x30100001					\$t140	141	0
	4194872	0x30100001					\$t141	142	0
	4194876	0x30100001					\$t142	143	0
	4194880	0x30100001					\$t143	144	0
	4194884	0x30100001					\$t144	145	0
	4194888	0x30100001					\$t145	146	0
	4194892	0x30100001					\$t146	147	0
	4194896	0x30100001					\$t147	148	0
	4194900	0x30100001					\$t148	149	0
	4194904	0x30100001					\$t149	150	0
	4194908	0x30100001					\$t150	151	0
	4194912	0x30100001					\$t151	152	0
	4194916	0x30100001					\$t152	153	0
	4194920	0x30100001					\$t153	154	0
	4194924	0x30100001					\$t154	155	0
	4194928	0x30100001					\$t155	156	0
	4194932	0x30100001					\$t156	157	0
	4194936	0x30100001					\$t157	158	0
	4194940	0x30100001					\$t158	159	0
	4194944	0x30100001					\$t159	160	0</