Bahria University,

Karachi Campus



LAB EXPERIMENT NO.

\_\_\_09\_\_\_\_

LIST OF TASKS

|  |  |
| --- | --- |
| TASK NO | OBJECTIVE |
| QUESTION NO.1 | **Execute above mention program and analyze outputs.** |
| QUESTION NO.2 | **Performing Bitwise OR Instruction with Mask 00000001** |
| QUESTION NO.3 | **Write a program which perform bitwise inversion using XOR function i.e input value = 0, output value = 1** |
| QUESTION NO.4 |  |
| QUESTION NO.5 |  |
| QUESTION NO.6 |  |
| QUESTION NO.7 |  |
| QUESTION NO.8 |  |
| QUESTION NO.9 |  |
| QUESTION NO. |  |
| QUESTION NO. |  |
| QUESTION NO. |  |
| QUESTION NO. |  |
| QUESTION NO. |  |
| QUESTION NO. |  |

Submitted On:

\_\_\_12/ 12/2019\_\_\_

(Date: DD/MM/YY)

**QUESTION-1: Execute above mention program and analyze outputs.**

**Code:**

#Performing Bitwise AND Instruction with Mask 11111110

.data

newLine: .asciiz "\n"

.text

main:

li $a1,10

jal showNumber # jump and link

li $a1,10

jal clearBitZero

move $a1,$v0

jal showNumber

li $v0,10

syscall

# Expecta a number in a1

showNumber:

la $a0,newLine

li $v0,4

syscall

move $a0,$a1

li $v0,1

syscall

jr $ra # return address

clearBitZero:

# Mask a mask

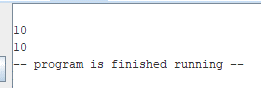
li $s0,-1

sll $s0,$s0,1 # Shift left logical shifts a register value left by the shift amount listed in the instruction

and $v0,$a1,$s0

jr $ra

**Output:**



**QUESTION-2: Performing Bitwise OR Instruction with Mask 00000001**

**Code:**

.data

newLine: .asciiz "\n"

.text

main:

li $a1,6

jal showNumber

li $a1,6

jal AddBitOne

move $a1,$v0

jal showNumber

li $v0,10

syscall

# Expecta a number in a1

showNumber:

la $a0,newLine

li $v0,4

syscall

move $a0,$a1

li $v0,1

syscall

jr $ra

AddBitOne:

# Mask a mask

li $s0,1

or $v0,$a1,$s0

jr $ra

**OUTPUT:**



**QUESTION-3: Write a program which perform bitwise inversion using XOR function i.e input value = 0, output value = 1**

**Code:**

.data

in: .asciiz"Enter Input = \n"

out:.asciiz"Output After Applying XOR gate =\n"

.text

la $a0,in

li $v0,4

syscall

li $v0,5

syscall

xori $t0,$v0,1

la $a0,out

li $v0,4

syscall

move $a0,$t0

li $v0,1

syscall

li $v0,10

syscall

**OUTPUT:**

