**Bahria University, Lahore Campus**

Department of Computer Sciences

Lab Journal 08

**(Spring 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Computer Architecture & Organization Lab** |  |
| Course Code: | CEL 221 | Max Marks: 15 |
| Faculty’s Name: | Maryam Munawar | Lab Engineer: |

Name: \_AFFAN AHMAD \_\_ Enroll No: \_03-134221-003\_\_\_\_\_\_\_

## Lab Tasks:

### Task 1: 5 Marks

Implement a program to prompt the user for two numbers, the first being any number and the second a prime number. Return to the user a 0 if the second number is a prime factor for the first, or any number if it is not. For example, if the user enter 60 and 5, the program returns 0. If the user enters 62 and 5, the program returns 2.

.data

a:.asciiz"enter the first number :"

b:.asciiz"enter the prime number number :"

r:.asciiz"0"

r2:.asciiz""

.text

addi $t2,$zero,2

li $v0,4

la $a0,a

syscall

li $v0 5

syscall

move $t0,$v0

li $v0,4

la $a0,b

syscall

li $v0 5

syscall

move $t1,$v0

while :

div $t0,$t0,$t2

mfhi $t3

beq $t3,0,val

add $t2,$t2,1

j while

val:

beq $t2,$t1,print

print :

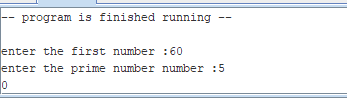
li $v0 1

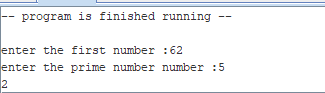
move $a0 ,$t3

syscall

li $v0, 10

syscall





**Task 2: 5 Marks**

Implement an overflow check for multiplication where the two numbers being multiplied are both always positive. Why is this simpler than the generic check implement in the mulo operator?

**.data**

**a:.asciiz"enter the first number :"**

**b:.asciiz"enter the second number :"**

**r:.asciiz"your number is negative "**

**result :.asciiz"your result is :"**

**r2:.asciiz""**

**x:.word 10**

**.text**

**main:**

**li $v0 ,4**

**la $a0 ,a**

**syscall**

**li $v0 ,5**

**syscall**

**move $t0,$v0**

**blt $t0,0,check**

**li $v0 ,4**

**la $a0 ,b**

**syscall**

**li $v0 ,5**

**syscall**

**move $t1,$v0**

**blt $t1,0,check**

**mult $t1,$t0**

**mflo $t3**

**li $v0 ,4**

**la $a0 ,result**

**syscall**

**li $v0 1**

**move $a0, $t3**

**syscall**

**li $v0 10**

**syscall**

**check :**

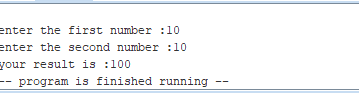
**li $v0 ,4**

**la $a0 ,r**

**syscall**

**li $v0 10**

**syscall**



**Task 3: 5 Marks**

Implement a program which multiplies a user input by 10 using only bit shift operations and addition. Check to see if your program is correct by using the mult and mflo operators. Your program should include a proper and useful prompt for input, and print the results in a meaningful manner.

.data

a:.asciiz"enter the first number :"

b:.asciiz"enter the prime number number :"

r:.asciiz"0"

result :.asciiz"your result is :"

r2:.asciiz""

x:.word 10

.text

addi $t0, $zero ,10

#mflo $t0

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $t1,$v0

li $v0, 4

la $a0, result

syscall

mult $t1,$t0

mflo $t3

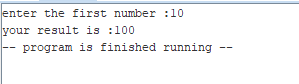
li $v0,1

move $a0, $t3

syscall

li $v0, 10

syscall



**Task 4: 5 Marks**

Implement a program which multiplies a user input by 15 using only bit shift operations and addition. Check to see if your program is correct by using the mult and mflo operators. Your program should include a proper and useful prompt for input, and print the results in a meaningful manner.

.data

a:.asciiz"enter the first number :"

b:.asciiz"enter the prime number number :"

r:.asciiz"0"

result :.asciiz"your result is :"

r2:.asciiz""

x:.word 10

.text

addi $t0,$zero,15

#mflo $t0

li $v0, 4

la $a0, a

syscall

li $v0 ,5

syscall

move $t1,$v0

li $v0, 4

la $a0, result

syscall

mult $t1,$t0

mflo $t3

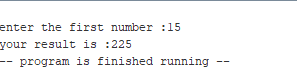
li $v0,1

move $a0, $t3

syscall

li $v0, 10

syscall



**Lab Grading Sheet :**

|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Max Marks** | **Obtained Marks** | **Comments(*if any*)** |
| a. | 5 |  |  |
| b. | 5 |  |  |
| c. | 5 |  |  |
| d | 5 |  |  |
| **Total** | **20** |  | **Signature** |

**Note : Attempt all tasks and get them checked by your Instructor**