

Objective of the Lab/Program

The objective of this program is to find the prime numbers between a range of number inclusive of the input.

Assembly Source Codes

```
#program to calculate sum of numbers
.data
A : .asciiz "\nPlease enter A(Lower Bound): "
B : .asciiz "\nPlease enter B(Upper Bound): "
str1 : .asciiz "\nPlease input: "
str2 : .asciiz "\nBoth A and B are the same value"
str3 : .asciiz "\nError: Number needs to be positive"
str4: .asciiz "\nNo Prime value exists in this range"
.text

#Storing variable A in register $s0
la $a0,A
li $v0, 4
syscall
li $v0,5
syscall
move $s0, $v0

#Storing variable B in register $s1
la $a0,B
li $v0, 4
syscall
li $v0,5
syscall
move $s1, $v0

#Check if negative
bltz $s0, negative
bltz $s1, negative

#Check if same
beq $s0, $s1, same

#variable definition
move $s3, $s0 #outer increment defined as low
li $s4, 2      #increment count
li $s5, 2      #Used to split in half
move $t1, $0
```

```
#Loop for primes
loop1:
div $s3, $s5
beq $s3, 1, noprime
li $s4, 2
mflo $s6 #Used as half point
    loop2:
    div $s3, $s4
    mfhi $s7

    beqz $s7, noprime #if remainder = 0 exit
    beq $s4, $s6, prime #prints if prime
    addi $s4, $s4, 1
    j loop2
noprime:
beq $s6, 1, prime #specail case where the mid point is 1, only runs for when looking at 2
and 3
beq $s3, $s1, endnopr
addi $s3, $s3, 1
j loop1

prime:
beq $s3, $s1, endpr
move $a0, $s3
li $t1, 1
li $v0, 1    #print, including syscall
syscall
addi $s3, $s3, 1
j loop1

endpr:
move $a0, $s3
li $t1, 1 #boolean for if prime exists in range
li $v0, 1
syscall
li $v0, 10 #terminate program
syscall

endnopr:
beqz $t1, nevpr
end:
li $v0, 10 #terminate program
syscall
```

```
negative:
la $a0,str3
li $v0, 4
syscall
li $v0, 5
li $v0, 10
syscall
```

```
same:
la $a0,str2
li $v0, 4
syscall
li $v0, 10
syscall
```

```
nevpr:
la $a0,str4
li $v0, 4
syscall
li $v0, 10
syscall
```

Screen shot of the results

```
Please enter A(Lower Bound): 1
Please enter B(Upper Bound): 1

Both A and B are the same value
-- program is finished running --
```

```
Please enter A(Lower Bound): 1
Please enter B(Upper Bound): 10
2357
-- program is finished running --
```

```
Please enter A(Lower Bound): -2
Please enter B(Upper Bound): 20

Error: Number needs to be positive
-- program is finished running --
```

```
Please enter A(Lower Bound): 24
Please enter B(Upper Bound): 26

No Prime value exists in this range
-- program is finished running --
```

Conclusion and References

After nesting two loops the program is able to check all the prime numbers inside a range. It also has escape cases which stop the program if: there are no prime numbers in the range, one input is negative and both inputs are the same.