|  |  |
| --- | --- |
|  | **Computer Organization & Assembly Language**  **BSCS 3rd**  **Department of Computer Science**  **Bahria University, Lahore Campus** |

**Quiz: 2**

Date: Week 6, 30th March 2023

Name: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Roll No: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

|  |  |  |  |
| --- | --- | --- | --- |
| **Evaluation of CLO** | **Question Number** | **Marks** | **Obtained Marks** |
| **CLO1: CLO statement**  *Simulate the internal representation of data, and show how data is stored and accessed in, I/O modules, and the interconnecting components of the computer systems* | 1 | 2.5 |  |
|  |  |  |
| **Total Marks** | | **2.5** |  |

**Question 1.** Write a program to prompt the user for a number, and determine if that number is prime. Your program should print out "Number n is prime" if the number is prime, and "Number n is not prime if the number is not prime. The user should be able to enter input a "-1" is entered. It should print an error if 0, 1, 2 or any negative number other than -1 are entered.

Note: Purpose: To find all primes from 3 to n

Registers: $s0 - The largest number to check if prime

$s1 - The loop counter over the primes (outer loop)

$s2 - The largest value to use to check if prime ($s1/2)

$s3 - The loop counter to check if prime

**Sol**

.data

prompt: .asciiz "Enter a number (-1 to exit): "

prime\_msg: .asciiz "Number %d is prime\n"

not\_prime\_msg: .asciiz "Number %d is not prime\n"

error\_msg: .asciiz "Error: Invalid input\n"

.text

.globl main

main:

li $s0, 0 # Initialize the largest number to check if prime

input\_loop:

# Print the prompt

li $v0, 4

la $a0, prompt

syscall

# Read the input number

li $v0, 5

syscall

move $s0, $v0 # Save the input number in $s0

# Exit the program if -1 is entered

beq $s0, -1, exit\_program

# Check for invalid input

blt $s0, -1, print\_error

beq $s0, 0, print\_error

beq $s0, 1, print\_error

# Initialize the loop counter over the primes

li $s1, 3

prime\_check:

# Initialize the largest value to use to check if prime

move $s2, $s1

div $s2, $s2, 2

# Initialize the loop counter to check if prime

li $s3, 2

prime\_inner\_check:

rem $t0, $s1, $s3

beqz $t0, not\_prime

addi $s3, $s3, 1

blt $s3, $s2, prime\_inner\_check

# The number is prime

li $v0, 1

move $a0, $s1

syscall

la $a0, prime\_msg

li $v0, 4

syscall

addi $s1, $s1, 1

blt $s1, $s0, prime\_check

b input\_loop

exit\_program:

li $v0, 10

syscall

print\_error:

la $a0, error\_msg

li $v0, 4

syscall

b input\_loop

not\_prime:

la $a0, not\_prime\_msg

li $v0, 4

syscall

addi $s1, $s1, 1

blt $s1, $s0, prime\_check

b input\_loop