**Course Code: EE488**

**Assignment -4**

**PREPARED BY**

**Khandoker Samiul Hoque**

**Student Id: 19837**

**Question No 1 Answer**

**utils.asm**

# File: utils.asm

# Purpose: To define utilities which will be used in MIPS programs.

#

# Subprograms Index:

# Exit - Call syscall with a server 10 to exit the program

# NewLine - Print a new line character (\n) to the console

# PrintInt - Print a string with an integer to the console

# PrintString - Print a string to the console

# PromptInt - Prompt the user to enter an integer, and return

# it to the calling program.

#

# subprogram: PrintNewLine

# purpose: to output a new line to the user console

# input: None

# output: None

# side effects: A new line character is printed to the

# user's console

.data

\_\_PNL\_newline: .asciiz "\n"

.text

PrintNewLine:

li $v0, 4

la $a0, \_\_PNL\_newline

syscall

jr $ra

# subprogram: PrintInt

# purpose: To print a string to the console

# input: $a0 - The address of the string to print.

# $a1 - The value of the int to print

# returns: None

# side effects: The String is printed followed by the integer value.

PrintInt:

# Print string. The string address is already in $a0

li $v0, 4

syscall

# Print integer. The integer value is in $a1, and must

# be first moved to $a0.

move $a0, $a1

li $v0, 1

syscall

#return

jr $ra

# subprogram: PromptInt

# purpose: To print the user for an integer input, and

# to return that input value to the caller.

# input: $a0 - The address of the string to print.

# returns: $v0 - The value the user entered

# side effects: The String is printed followed by the integer value.

PromptInt:

# Print the prompt, which is already in $a0

li $v0, 4

syscall

# Read the integer value. Note that at the end of the

# syscall the value is already in $v0, so there is no

# need to move it anywhere.

move $a0, $a1

li $v0, 5

syscall

#return

jr $ra

# subprogram: PrintString

# purpose: To print a string to the console

# input: $a0 - The address of the string to print.

# returns: None

# side effects: The String is printed to the console.

PrintString:

addi $v0, $zero, 4

syscall

jr $ra

# subprogram: Exit

# purpose: to use syscall service 10 to exit a program

# input: None

# output: None

# side effects: The program is exited

Exit:

li $v0, 10

syscall

**a.**

.data

prompt: .asciiz "Enter the number: "

Display\_result: .asciiz "Result = input\*10: "

.text

main:

la $a0, prompt

jal PromptInt

move $t0, $v0

sll $t1, $t0, 3

sll $t2, $t0, 1

add $t3, $t1, $t2

la $a0, Display\_result

move $a1, $t3

jal PrintInt

jal Exit

.include "utils.asm"

**b**.

.data

input: .space 4

input\_Size: .word 4

prompt: .asciiz "Enter the word: "

result: .asciiz "Result: "

.text

main:

la $a0, prompt

jal PrintString

li $v0, 8

la $a0, input

lw $a1, input\_Size

syscall

jal PrintNewLine

li $v0, 4

li $t0, 0

loop:

lb $t1, input($t0)

beq $t1, 0, print

blt $t1, 'a', again

bgt $t1, 'z', again

sub $t1, $t1, 32

sb $t1, input($t0)

again:

addi $t0, $t0, 1

j loop

print:

la $a0, result

jal PrintString

la $a0, input

jal PrintString

jal Exit

.include "utils.asm"

**c.**

.data

input: .space 4

inputSize: .word 4

prompt: .asciiz "Enter the 32-bits input char: "

result: .asciiz "Result: "

.text

main:

la $a0, prompt

jal PrintString

li $v0, 8

la $a0, input

lw $a1, inputSize

syscall

jal PrintNewLine

li $v0, 4

li $t0, 0

loop:

lb $t1, input($t0)

beq $t1, 0, print

blt $t1, 'A' case

bgt $t1, 'Z', case

add $t1, $t1, 32

sb $t1, input($t0)

case:

addi $t0, $t0, 1

j loop

print:

la $a0, result

jal PrintString

la $a0, input

jal PrintString

jal Exit

.include "utils.asm"

**Question No 2 Answer**

.data

input: .asciiz "input the number: "

output: .asciiz " The prime numbers from 3 to input are: "

.text

main:

li $v0, 4

la $a0, input

syscall

li $v0, 5

syscall

move $t0, $v0

li $v0, 4

la $a0, output

syscall

li $t1, 3

li $t2, 2

loop:

div $t1, $t2

mfhi $t3

beq $t3, $0, not\_prime\_no

addi $t2, $t2, 1

blt $t2, $t1, loop

li $v0, 1

move $a0, $t1

syscall

li $v0, 4

la $a0, space

syscall

not\_prime\_no:

addi $t1, $t1, 1

li $t2, 2

blt $t1, $t0, loop

li $v0, 10

syscall

.data

space: .asciiz " "

**Question No 3 Answer**

.data

prompt: .asciiz "Input a number from 3 to 100: "

newline: .asciiz "\n"

space: .asciiz " "

output: .asciiz "The Prime factors of your input are: "

.text

.globl main

main:

li $v0, 4

la $a0, prompt

syscall

li $v0, 5

syscall

move $t0, $v0

li $v0, 4

la $a0, newline

syscall

li $v0, 4

la $a0, output

syscall

li $t1, 2

factor\_loop:

ble $t1, $t0, factor\_check # if divisor <= number, check if it's a factor

j end # if divisor > number, end program

factor\_check:

div $t0, $t1 # divide number by divisor

mfhi $t2 # get the remainder

beq $t2, $zero, factor\_output # if remainder is 0, divisor is a factor

addi $t1, $t1, 1 # if remainder is not 0, increment divisor

j factor\_loop

factor\_output:

li $v0, 1

move $a0, $t1

syscall

li $v0, 4

la $a0, space

syscall

div $t0, $t1

mflo $t0

j factor\_loop

end:

li $v0, 4

la $a0, newline

syscall

li $v0, 10

syscall

**Question No 4 Answer**

.data

prompt: .asciiz "Enter an integer: "

even\_display: .asciiz "The number is even\n"

odd\_display: .asciiz "The number is odd\n"

.text

.globl main

main:

li $v0, 4

la $a0, prompt

syscall

li $v0, 5

syscall

move $s0, $v0

# Check if number is even or odd

sll $t0, $s0, 31

srl $t0, $t0, 31

xor $s1, $s0, $t0

srl $t1, $s1, 1

sll $t2, $t1, 1

bne $s0, $t2, odd

li $v0, 4

la $a0, even\_display

syscall

j end

odd:

li $v0, 4

la $a0, odd\_display

syscall

end:

li $v0, 10

syscall

**Question No 5 Answer**

.data

quarter: .word 25

dime: .word 10

nickel: .word 5

quarter\_no: .asciiz " quarter(s), "

dime\_no: .asciiz " dime(s), "

nickel\_no: .asciiz " nickel(s), "

penny\_no: .asciiz " penny(s) \n"

prompt: .asciiz "Enter a number in range 0-100: "

.text

li $v0, 4

la $a0, prompt

syscall

li $v0, 5

syscall

move $t0, $v0

lw $t1, quarter

div $t0, $t1

mflo $t2

mfhi $t0

lw $t1, dime

div $t0, $t1

mflo $t3

mfhi $t0

lw $t1, nickel

div $t0, $t1

mflo $t4

mfhi $t0

li $v0, 1

move $a0, $t2

syscall

li $v0, 4

la $a0, quarter\_no

syscall

li $v0, 1

move $a0, $t3

syscall

li $v0, 4

la $a0, dime\_no

syscall

li $v0, 1

move $a0, $t4

syscall

li $v0, 4

la $a0, nickel\_no

syscall

li $v0, 1

move $a0, $t0

syscall

li $v0, 4

la $a0, penny\_no

syscall

li $v0, 10

syscall