Assignment 3

Adarsh Tiwari (B21ES002)

September 2023

1 Question 1

1.1 Part a

```
.data

input: .string "Hello World!"


.text
.globl main

print_string:
    addi sp sp -16
    sw ra 0(sp)
    sw a0 4(sp)
    sw a1 8(sp)      #what if we know that we have only one input argument

    add t0 x0 a0
Loop:    lbu a1,0(t0)
         beq a1,x0,Return
         addi a0,x0,11
         ecall
         addi t0 t0 1
    j Loop

    lw ra 0(sp)
    lw a0 4(sp)
    lw a1 8(sp)
    addi sp sp 12


    Return:    ret

main:

    la a0 input
    jal ra print_string

    addi a0 x0 10
    ecall
```

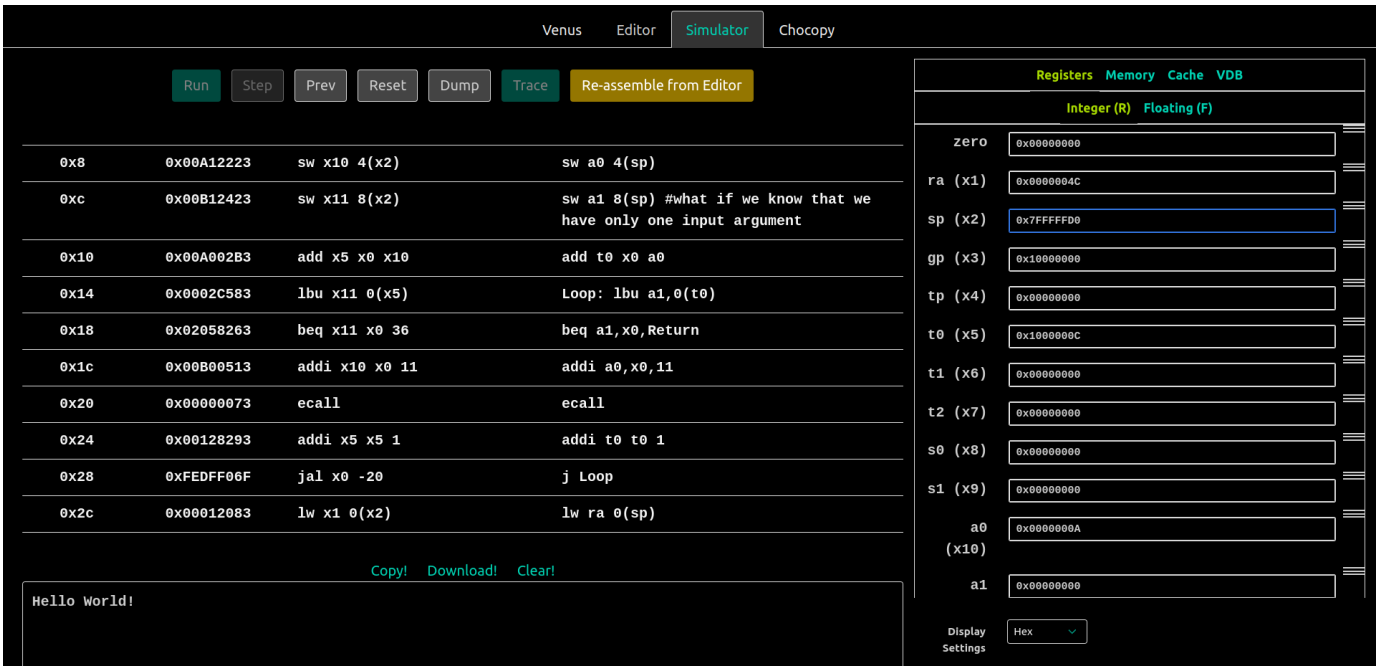

Figure 1: Output

1.2 Part b

```
.data
input: .string "Hello World!"

.text
.globl main

print_string:
    addi sp sp -16
    sw ra 0(sp)
    sw a0 4(sp)
    sw a1 8(sp)

    add t0 x0 a1   # copy the length of string
    add t1 x0 a0    # copy the base address of string
    Loop: beq t0 x0 Return
        addi t0 t0 -1
        addi a0 x0 11
        lbu a1 0(t1)
        ecall
        addi t1 t1 1
        j Loop


Return:    lw ra 0(sp)
    lw a0 4(sp)
    lw a1 8(sp)
    addi sp sp +16
    ret

main:
    la a0 input
    addi a1 x0 12
    jal ra print_string

    addi a0 x0 10
    ecall
```

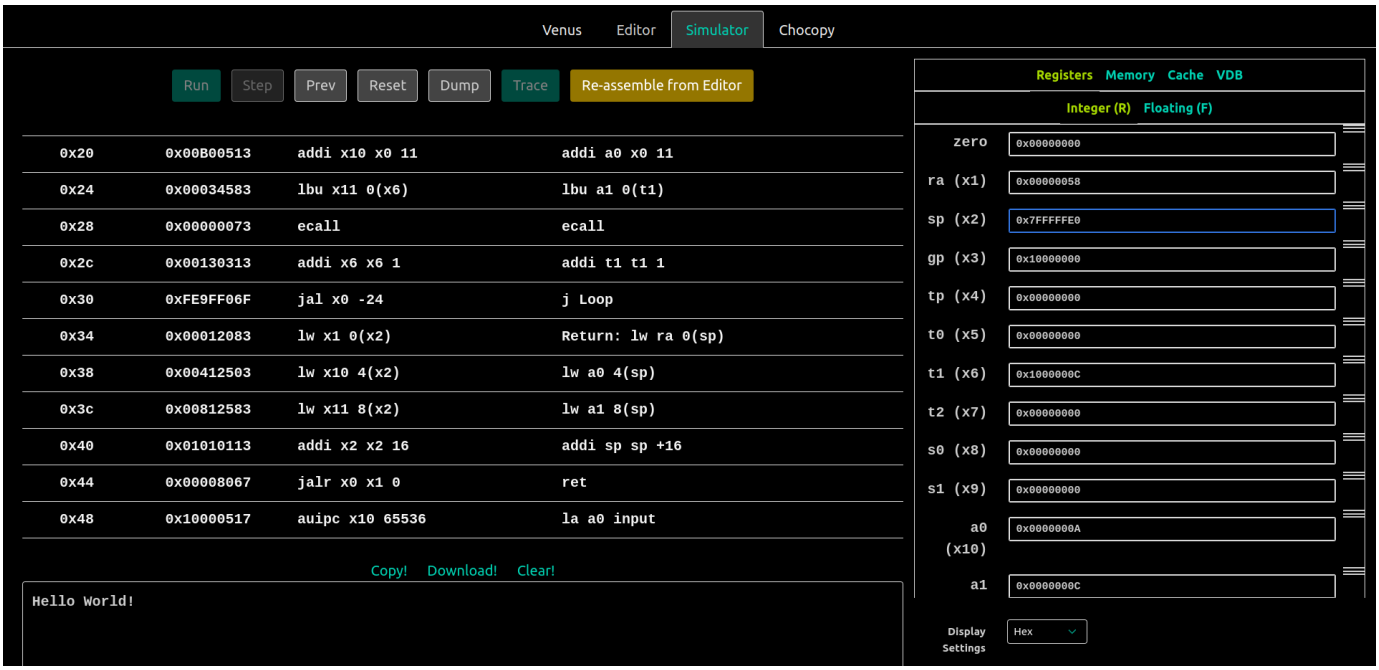

Figure 2: Output

1.3 Part c

```
.data
input: .string "Hello World!"

.text
.globl main

print_string:
    addi sp sp -16
    sw ra 0(sp)
    sw a0 4(sp)
    sw a1 8(sp)

    add t1 x0 a0    # copy the base address of string
    lbu t0 0(t1)
    beq t0 x0 Return
    Loop:
        lbu a1 0(t1)
        addi a0 x0 11
        ecall
        addi t1 t1 1
        beq a1 x0 Return
        j Loop


Return:    lw ra 0(sp)
    lw a0 4(sp)
    lw a1 8(sp)
    addi sp sp +16
    ret

main:

    la a0 input
    jal ra print_string

    addi a0 x0 10
    ecall
```

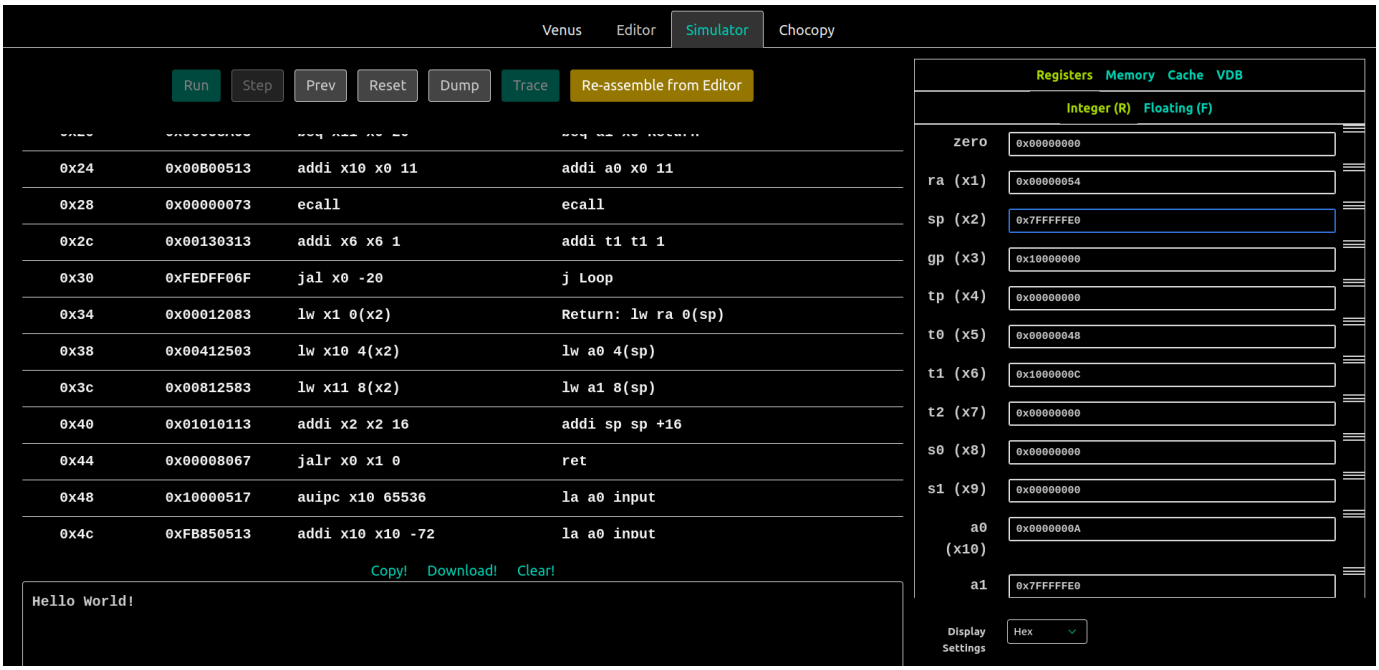

Figure 3: Output

2 Question 3

```
.text
.globl main

next_half_int:

    addi a0 x0 0
    addi a0 a0 1

    addi t0 t0 1
    slli t0 t0 15

    slt t1 a0 t0
    bne t1 x0 True1
    addi a0 x0 0

True1:    slli t0 t0 1
    sub t0 x0 t0

    slt t1 t0 a0

    bne t1 x0 True2
    addi a0 x0 0

True2:    jr ra



main:
    jal next_half_int

    add a1 a0 x0
    addi a0 x0 1

    ecall

    addi a0 x0 10

    ecall
```

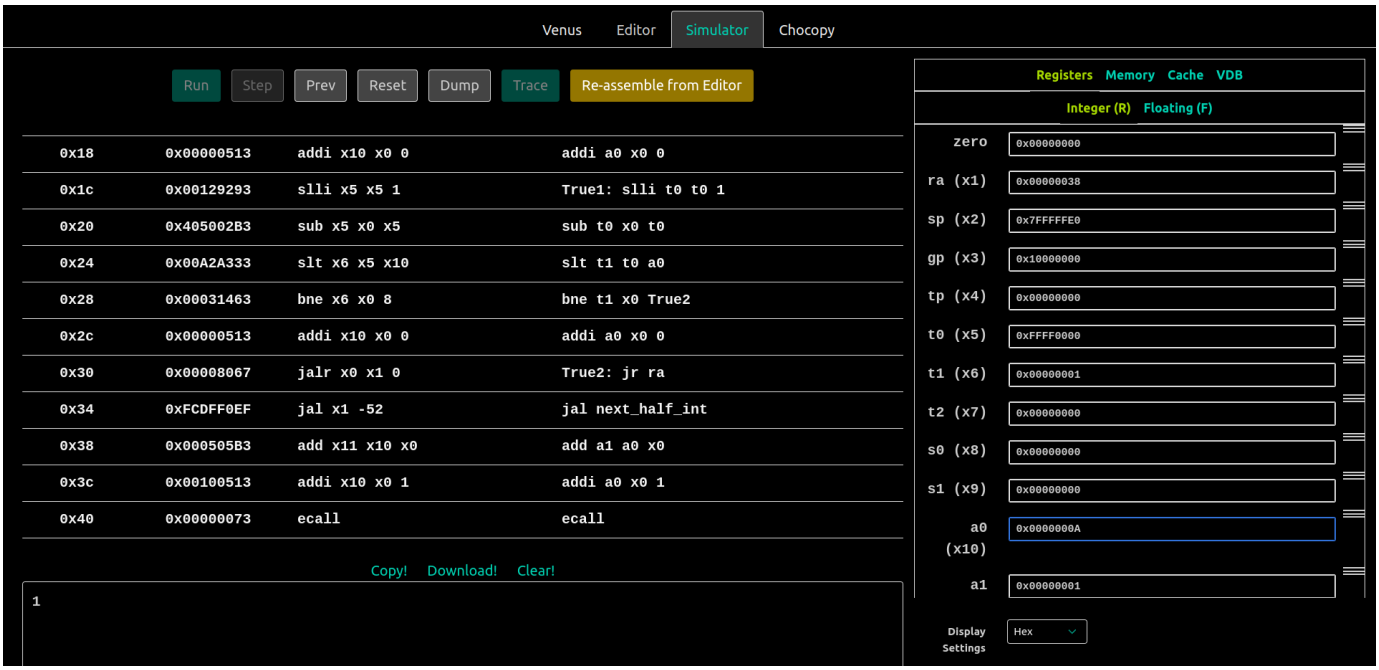

Figure 4: Output

3 Question 5

```
.data
neg3:   .asciiz "f(-3) is 11, and it is: "
neg2:   .asciiz "f(-2) is 21, and it is: "
neg1:   .asciiz "f(-1) is 45, and it is: "
zero:   .asciiz "f(0) is -3, and it is: "
pos1:   .asciiz "f(1) is 12, and it is: "
pos2:   .asciiz "f(2) is 62, and it is: "
pos3:   .asciiz "f(3) is 16, and it is: "

output: .word   11, 21, 45, -3, 12, 62, 16
.text
main:
        la      a0, neg3
        jal     print_str
        li      a0, -3
        jal     f               # f(-3); is 11
        jal     print_int
        jal     newline

        la      a0, neg2
        jal     print_str
        li      a0, -2
        jal     f               # f(-2); is 21
        jal     print_int
        jal     newline

        la      a0, neg1
        jal     print_str
        li      a0, -1
        jal     f               # f(-1); is 45
        jal     print_int
        jal     newline

        la      a0, zero
        jal     print_str
        li      a0, 0
        jal     f               # f(0); is -3
        jal     print_int
        jal     newline

        la      a0, pos1
        jal     print_str
        li      a0, 1
        jal     f               # f(1); is 12
        jal     print_int
        jal     newline

        la      a0, pos2
        jal     print_str
        li      a0, 2
        jal     f               # f(2); is 62
        jal     print_int
```

```
        jal     newline

        la      a0, pos3
        jal     print_str
        li      a0, 3
        jal     f               # evaluate C(4,0); is 16
        jal     print_int
        jal     newline

        li      a0, 10
        ecall

# calculate f(a0)
f:
        la      t0, output      # Why is this a good idea?

        addi a0 a0 3
        slli a0 a0 2

        add t0 t0 a0
        lw a0 0(t0)

        # YOUR CODE

        jr      ra              # Remember to jr ra after your function!

print_int:
        mv a1, a0
        li      a0, 1
        ecall
        jr      ra

print_str:
        mv a1, a0
        li      a0, 4
        ecall
        jr      ra

newline:
        li      a1, '\n'
        li      a0, 11
        ecall
        jr      ra
```

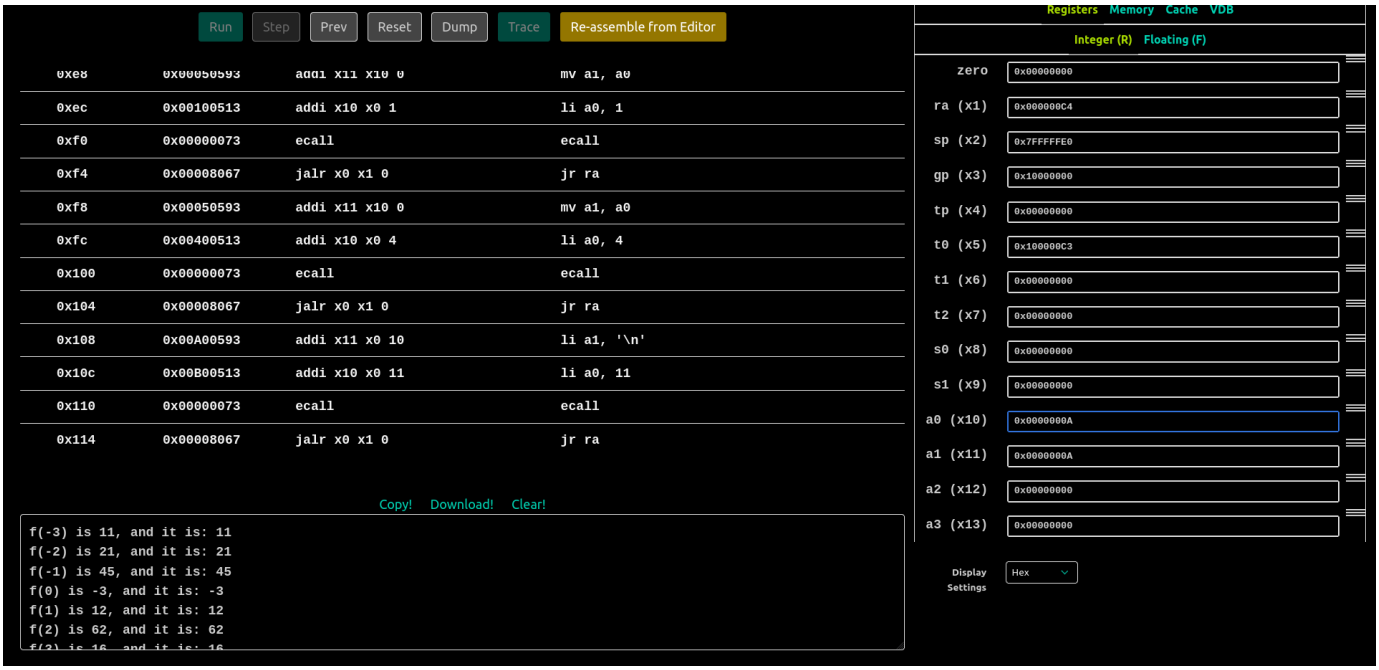

Figure 5: Output