

COMP1047 Lab Week 04

1. Work out the 2's complement representation for the following decimal numbers by hand.

(a) 45 (b) -130

Then, write a MIPS program to load the numbers above into registers \$s0 and \$s1 as unsigned numbers. You can place the values directly in the data memory segment and then use `lw` instruction to load them into registers. For example, the following program stores two integers $0000000A_{16}$ and 10000000_{16} in the data segment of the memory and then loads the first integer in \$s0 using `lw` instruction.

```
.data
int: .word 0x0000000A 0x10000000
.text
.globl main
main:
    la $t0, int          #load the base address
    lw $s0, 0($t0)       #load the first integer into $s0
```

Note that here we use assembler directive `.word`. You can find more assembler directives from the MIPS reference card. Now print out both numbers to the console using the `syscall` function. Check the output to see whether it is expected.

2. Implement the following C functions using MIPS procedure.

```
int non_leaf (int g, int h, int i, int j){
    int f;
    f = leaf (g+h, i+j);
    return f;
}

int leaf (int m, int n){
    int f;
    f = m-n;
    return f;
}
```

3. Write an MIPS program that reads a string from console and then print out the string in its reverse alphabetical order. For example, if the string from user is \Hello", then you should print out \olleH".

Hint: To read a string from user, you need to allocate a memory buffer (.data space) of appropriate sizes using `.space` directive. For example, the following statement requests 10-byte space of memory space with the starting address as `buffer`.

```
        .data
buffer:  .space 10
```

The following segment reads a string from console. At the end of the syscall, the string is stored in the buffer in data segment.

```
la $a0, buffer    #buffer address to $a0
li $a1, 10        #string length to $a1
li $v0, 8         #read string
syscall
```

4. For the above question, instead of printing out the reverse order of the string, please change the third character of the string to upper case (assuming it was typed in as lower case) and then print out the string. For example, if the input is \Hello", then change it to \HeLlo" before printing out. Note, you can only use `lw` and `sw` instructions for data transfer to/from the main memory.