Rules for Submitting the Assignment
You should submit your assignment through blackboard, as 2 files, with the Subjects "Home Work 2, Part 1" and "Home Work 2, Part 2". The files should be named according to the student names. For example, if Bhuvan Middha is submitting then his files should be Bhuvan-Middha.s and Bhuvan-Middha.txt. The first file (the program) program must be in plain ASCII (i.e., a text file and not a .doc file). Clearly state in the file (as a comment) your team partner's name if you have one.

Part 1
Implement the following C program in the MIPS-I assembly language: Note that for translating the printf() and scanf() library calls, you have to use SPIM's API. Comment each instruction of your code by specifying which high-level language instruction it corresponds to.

```c
struct {
    int forward_length;
    int reverse_length;
} record;

main()
{
    char str[30];

    printf("Please type a word: ");
    scanf("%s", str);

    printf("You typed the word: ");
    record.forward_length = print_string(str, 0);
    printf("; that was a %d letter word!\n", record.forward_length);
    printf("In reverse, that's: ");
    record.reverse_length = print_string(str, 2);
    printf("; it's still got %d letters.\n", record.reverse_length);
}

print_string(char *str, int direction)
{
    int length;

    if (*str == 0)
        return 0;
```
if (direction == 0)
{
    printf("%c", *str);
    length = print_string(str+1, direction);
}
else
{
    length = print_string(str+1, direction);
    printf("%c", *str);
}
return (length + 1);

NOTE:

1. You should execute your assembly programs on a SPIM simulator. If SPIM is unable to
assemble your program, you will lose at least 50% of the grade. xspim is available in the
Glue machines as well as in the wam machines. PCSpim can be freely downloaded from the
internet.

2. You should specify registers by their symbolic names (such as $a0 and $a1, and not $4 and
$5). Similarly, memory locations should be specified only by labels, and not the absolute
addresses.

3. You should use the MIPS-I AL register conventions: Registers $a0-$a3 are used for passing
subroutine parameters, and register $v0 is used for passing the subroutine return value, $sp
is used for the stack pointer, $ra is used for storing the subroutine return address, and
subroutines preserve all the caller’s registers (except for the $t registers) by storing them on
the stack frame.

Part 2

1. What happens when a load instruction is executed in MIPS-I? (a) The bits of the register are
set to all ones.
(b) The bit pattern in the register is copied to a location in memory.
(c) A bit pattern at a memory location is copied to the register. The memory is set to all zeros.
(d) A bit pattern at a memory location is copied to the register. Memory is not changed.

2. By convention, the instructions of a program are placed in a designated section of memory.
What is this section called? (a) Data section. (b) Stack section. (c) Program section. (d) Text
section.

3. What general purpose register of MIPS-I ALA is permanently set to zero? (a) $0  (b) $1  (c)
$31  (d) $32

4. Where do the operands for an arithmetic instruction come from in the MIPS-I ALA? (a) Both
operands are registers. (b) Both operands come from memory. (c) One operands must be a
register, the other one may be memory or a register. (d) One operand must be a register, the
other may be a register or may be part of the machine instruction.

5. What is an opcode? (a) The part of an instruction that designates the data to be used. (b)
The part of an instruction that designates an operation to be performed. (c) The part of the
processor chip that performs decoding operations. (d) The part of an instruction that is used as
data in an operation.

6. What is the mnemonic name of a register? (a) A register number like $0 or $31. (b) A name
that helps you remember the hardware characteristics of the register. (c) The bit pattern that
designates the register in a machine instruction. (d) A name like $s0 that helps you remember
the conventional software uses for the register.

7. What is an assembler directive, such as the directive .text? (a) An assembly language statement
that results in one machine language instruction. (b) One of the menu choices in the SPIM menu
system. (c) An instruction that causes an operation on data. (d) A statement that tells the
assembler something about what the programmer wants, but does not itself result in any machine
instructions.

8. What is a symbolic address? (a) A location in memory containing symbolic data. (b) A byte
in memory that holds the address of data. (c) The symbol given as the argument for a directive.
(d) A name used in source code for a location in memory.

9. At what address does the SPIM simulator put the first machine instruction when it is running
with the Bare Machine option turned ON? (a) 0x00000000 (b) 0x00400000 (c) 0x10000000 (d) 0xFFFFFFFF

10. Consider a label label1 declared with a .word directive in the .data section of an assembly
language program. A memory address is assigned to this label
(a) by the assembler
(b) by the linker
(c) by the loader
(d) by the run-time hardware
(e) by the operating system
(f) none of the above

11. Consider a global variable int var1; declared at the top of a C program. An absolute
memory address (i.e., the exact address occupied by this variable in memory) is assigned to this
variable
(a) by the compiler (C program → assembly language translator)
(b) by the assembler
(c) by the run-time hardware
(d) by the operating system
(e) none of the above

12. Consider a local variable int var2; declared within a function in a C program. An absolute
memory address (i.e., the exact address occupied by this variable in memory) is assigned to this
variable
(a) by the compiler (C program → assembly language translator)
(b) by the assembler
(c) by the run-time hardware
(d) by the operating system
(e) none of the above