**Q No:1 The following problem deals with translating from C to MIPS. Assume that the variables g, h, i and j are given and could be considered 32-bit integers declared in C program**

**a.** f= g+h+i+j.

**b.** f= g+ (h+5).

**1.1** For the C statements above, what is the corresponding MIPS assembly code ? Use a

minimal number of MIPS assembly instructions.

**1.2** If the variables f, g, h, i and j have values 1, 2, 3, 4 and 5 respectively, what is the end

value of f?

**1.3** How many MIPS instructions are required to run these statements.

**Ans**: **1.1** The corresponding MIPS assembly code for the given C statement would be:

1. lw $t0, g

lw $t1, h

lw $t2, i

lw $t3, j

add $s0, $t0, $t1

add $s0, $s0, $t2

add $s0, $s $t3

1. lw $t0, g

lw $t1, h

addi $t2, $t1, 5

add $s0, $t0, $t2

**Ans:1.2** If the variables f,g,h,i and j have values 1,2,3,4,and 5,respectively,then the end value of f would be:

1. f =1 + 2 + 3 + 4 =10.
2. f=2 + (3 + 5) =10.

**Ans:1.3** For both statements, a total of 4 instructions are required, including the memory access instructions and arithmetic.

**Q2:**

**The following problem deals with translating from MIPS to C. Assume that the variables f,**

**g, h, i and j are assigned to registers $s0, $s1, $s2, $s3 and $s4 respectively. Assume that the**

**base address of the arrays A and B are in registers $s6 and $s7, respectively,**

**a.** add $s0, $s0, $s1

add $s0, $s0, $s2

add $s0, $s0, $s3

add $s0, $s0, $s4

**b.** lw $s0, 4($s6)

**2.1** For the MIPS assembly instruction above, what is the corresponding C statement?

**2.2** For the MIPS assembly instructions, rewrite the code to minimize the number of

instructions (if possible) needed to carry out the same functions.

**Ans: 2.1** For the MIPS assembly instruction “add $s0, $s0, $s1”, the corresponding C statement would be:

f = g + h;

For the remaining instruction in part (a).the corresponding C statement would be:

f = g + h + i+ j;

**2.2** The instruction sequence in part (a)can be simplified using the “addi ”instruction and “lw” instruction as follows:

**a.** add $s0, $s0, $s1 - > addi $s0, $s1, $s0

add $s0, $s0, $s2 - > addi $s0, $s2, $s0

add $s0, $s0, $s3 - > addi $s0, $s3, $s0

add $s0, $s0, $s4 - > addi $s0, $s4, $s0

**b.** lw $s0, 4($s6) lw $s0, 16($s6)

Note: The specific offset of 16 may vary depending on the size and data type of the elements in array A.