Sheet #1

1) Show the single MIPS instruction for the following C statements .

a)   X[10] = X[11] + c ;
b)   a = b + 100;

2) Translate the following hexadecimal code into MIPS instructions stating it's type.
   a) $(12345678)_{16}$
   b) $(BEADF00D)_{16}$

3) Add comments to the following MIPS code and describe in one sentence what it computes. Assume that $a0 is used for the input and initially contains n, a positive integer. Assume that $v0 is used for the output.

```
begin: addi  $t0, $zero, 0
       addi  $t1, $zero, 1
loop:  slt   $t2, $a0, $t1
       bne   $t2, $zero, finish
       add   $t0, $t0, $t1
       addi  $t1, $t1, 2
       j     loop
finish: add  $v0, $t0, $zero
```

4) The following program tries to copy words from the address in register $a0 to the address in the register $a1, counting the number of words copied in the register $v0. The program stops copying when it finds a word equal to 0. You do not have preserve the contents of the registers $v1, $a0, and $a1. This terminating word should be copied but not counted.

```
loop : lw    $v1, 0($a0)
       addi  $v0, $v0, 1
       sw    $v1, 0($a1)
       addi  $a0, $a0, 1
       addi  $a1, $a1, 1
       bne   $v1, $zero, loop
```

There are multiple bugs in this MIPS program; fix them and turn in a bug-free version.

4) Write MIPS code for the following statements:
>   a) if (i==j) then f=g+h
>   else f=g-h

>   b) while (x==k)
>   x=i+1

5) Implement the following section of C using MIPS assembly code:
>   for(i=0 ; i<x ; i=i+1)
>   y=y+i;

6) Consider the following fragment of C code:
>   for (i=0; i<=100; i=i+1)
>   {a[i] = b[i] + c;}

Assume that a and b are arrays of words and the base address of a is in $a0 and the base address of b is in $a1. Register $t0 is associated with variable i and register $s0 with c. Write the code for MIPS.

7) Write a program using MIPS code to
>   a) Get the sum of n numbers
>   b) Get the maximum of five numbers
>   c) Multiply any two positive integers X and Y