

Assignment # 2

- 1) If \$s1=0x21AC and \$s2=0x8C15, show the value of \$s3 in hex after execution of each of the following instruction below.

```
add    $s3, $s1, $s2
slt     $s3, $s1, $s2
sub     $s3, $s2, $s1
```

- 2) Find the minimum MIPS instruction to perform the following C statements. Use any registers for variables. Assume that the base register for array Y is in \$s0.

```
Y[10] = Y[3] + 8
a      = b - 5
k      = 3*m + 2
a++
```

- 3) For the following code segment write the machine language representation of each instruction in binary. The instruction codes are add->32, beq->4, addi->8, lw->35, j->2. Assume that Loop has the address of 0x4CB23

Loop:

```
beq    $t1, $t2, done
lw     $s1, 0($t0)
add    $s0, $s1, $s0
addi   $t1, $t1, 1
addi   $t0, $t0, 4
j      Loop
```

done:

- 4) Describe what would happen after main routine executes “jal sub1” and sub1(subroutine) executes “jr \$ra”.
-

Programming Question:

Write a program in MIPS assembly language that asks the users to enter a value for array size “n” and fill up the array with n integers. Then reverse the array and print it on the screen.

You are required to do the following in your program:

- 1) Create up to 80 bytes of space in your program
- 2) “n”, the array size, should be greater than zero and less than 20
- 3) Each element of the array should be positive and divisible by 3
- 4) Array should be reversed within itself. i.e. you must not create another array to place the elements in reverse order

You need to use following functions:

```
int main( )
{
    Begin: arraysize= readNum()
        Ok=verifySize(arraySize)
        if (ok==1)
            creatArray(array-size)
        else
            Go to Begin
        printArray(arraySize)
        reverseArray(arraySize)
        printArray (arraySize)
}
// -----
int readNum()
{
    input the arraySize
    return the arraySize to main program
}

// -----
int versifySize( int arraySize)
{
    if (0<arraySize< 20)
        return 1
    else
        return 0
}

// -----
```

```

void createArray(int arraySize)
{
    counter = 0
    making_array: arrayEntry = readNum()
        ok = checkNumPositive(arrayEntry)
        if (ok == 1)
            validNum = divisibleBy3(arrayEntry)
            if (validNum==1)
                insert arrayEntry into the array
                counter = counter + 1
            else
                go to making_array
        else
            go to making_array
    if (counter < arraySize)
        go to making_array
}

// -----
void reverseArray(int arraySize)
{
    head = first index of the array
    tail = last index of the array
    swap: if (head <=tail)
    {
        swap the content of head and tail (array[head] and array[tail] with each other
        increment the head value by one word
        decrement tail by one word
        go to swap
    }
}

// -----
void printArray( int arraySize)
{
    ..
    ..
}

// -----
int divisibleBy3(int arrayEntry)
{
    ...
    ...
}

```

```
}  
// -----  
int checkNumPositive(int arrayEntry)  
{  
...  
...  
}
```