Loops in MIPS

```
sum = 0;
for(i = 0; i < 100; i++){
        sum += a[i];
}
```

Replace the for statement using an if and goto statement:

```
sum =0;
i =0;
goto test;
loop: sum += a[i];
i++;
test: if (i < 100) goto loop;
```

Then in assembly:

```
                li $t2, 0                       # sum = 0
                or $t3, $0, $0                  # i = 0
                j test
Loop:   sll $t5, $t3, 2                 # temp = i * 4
                add $t5, $t5, $t4               # temp = temp + &a
                lw $t5, 0($t5)                  # load a[i] into temp
                add $t2, $t2, $t5               # sum += temp
                addi $t3, $t3, 1                # i++
Test:   slti $t5, $t3, 100              # test i < 100
                bne $t5, $zeor, loop            # if true, goto loop
```

For info about registry types: MIPS 3

```
while (save[i] == k){
        i+=1;
}
```

The solution within assembly will then be

```
# s3 <= i, s5 <= 5, *arr <= s0
LOOP:    SLL $t0, $s3, 2                  # t0 = i*4
                ADD $t0, $t0, $s0                # *t0 = arr + i
                LW  $t1, 0($t0)                  # t1 = arr[i]
                BNE $t1, $s5, DONE               # Loop done is not equal
                ADDI $s3, $s3, 1                 # i+=1
                j LOOP                           # jump back to loop again
DONE:
```

Example Program for finding largest value in an array

```
#largest.asm
# MIPS code to find the largest element in an array

        .text
        .globl main
main:   la      $s0,Arr         # address of arr *Arr
        la      $s1,size        # address of size
        lw      $s1,0($s1)      # s1 = 12 (size of array)
        ori     $t0,$0,1        # t0 = 1 (i=1)
        lw      $s2,0($s0)      # largest (s2) = arr[0]
LOOP:   beq     $t0,$s1,DONE    # if i == size, we're done
        sll     $t1,$t0,2       # t1 = i*4
        add     $t1,$s0,$t1     # t1 = &arr[i]
        lw      $t2,0($t1)      # t2 = arr[i]
        slt     $t3,$s2,$t2     # t3 =1 if largest < arr[i]
        beq     $t3,$0,NEXT     # if t3==0, then largest was bigger
        or      $s2,$t2,$0      # largest = arr[i]
NEXT:   addi    $t0,$t0,1       # i++
        j       LOOP            # back to condition check (if i<size)

DONE:   la      $a0,str         # Print str
        ori     $v0,$0,4        # 4 is syscall code for print string
        syscall

        or      $a0,$s2,$0      # print value in s2
        ori     $v0,$0,1        # 1 is sycall code for print int
        syscall
```

```
        ori     $v0,$0,10       # 10 is syscall code for end program
        syscall

        .data
Arr:    .word   17, 32, 21, -15, 99, 65, 42, 17, -80, 0, 19, 77
size:   .word   12
str:    .asciiz "The largest value is "
```