

```

# Simple routine to demo a loop
# Compute the sum of N integers: 1 + 2 + 3 + ... + N

.data
msg1: .asciiz "Number of integers (N)? "
msg2: .asciiz "Sum = "
lf:   .asciiz  "\n"

.text

.globl main
main:

    # Print msg1
    li    $v0,4          # print_string syscall code = 4
    la    $a0, msg1
    syscall

    # Get N from user and save
    li    $v0,5          # read_int syscall code = 5
    syscall
    move  $t0,$v0        # syscall results returned in $v0

    # Initialize registers
    li    $t1, 0         # initialize counter (i)
    li    $t2, 0         # initialize sum

    # Main loop body
loop: addi  $t1, $t1, 1    # i = i + 1
      add   $t2, $t2, $t1  # sum = sum + i
      beq   $t0, $t1, exit # if i = N, continue

```

```

        j      loop      # Jump to label loop (unconditional jump)

# Exit routine - print msg2
exit:  li      $v0, 4      # print_string syscall code = 4
        la      $a0, msg2
        syscall

# Print sum
li      $v0, 1      # print_string syscall code = 4
move    $a0, $t2
syscall

# Print newline
li      $v0, 4      # print_string syscall code = 4
la      $a0, lf
syscall

li      $v0, 10      # exit
syscall

```

```

=====
==

```

1. Convert the following c-like code into MIPS assembly code.

```
void swap (int v[], int k)
```

```
{int temp;
```

```
temp = v[k];
```

```
v[k] = v[k+1];
```

```
v[k+1] = temp;}
```

```
-----
```

2. Convert the following c-like code into MIPS assembly code.

```
if ( i == j )
```

```
    i++ ;
```

```
j-- ;
```

3.Convert the following c-like code into MIPS assembly code.

```
if ( i == j )
```

```
    i++ ;
```

```
else
```

```
    j-- ;
```

```
j += i ;
```

4. Convert the following c-like code into MIPS assembly code.

```
if ( i == j && i == k )
```

```
    j++ ;
```

```
    i-- ;
```

```
else
```

```
    j = i + k-2 ;
```

5. Convert the following c-like code into MIPS assembly code.

```
if ( i==j || i==k )
```

```
    i++ ;
```

```
    j-- ;
```

```
else
```

```
    j = i + k ;
```