Problem FC-1 (4 parts) **Branches, Jumps, Loops**

Part A Write a MIPS code fragment that jumps to instruction address 0x8A6B9C2C.

	`lui $1, 0x8A6B`	`# load 8A6B into upper 16 bits of $1`
	`ori $1, $1, 0x9C2C`	`# place 9C2C in the lower 16 bits`
	`jr $1`	`# jump to address in $1`

Part B Write a two instruction MIPS code fragment that branches to label `Target` if register $4 is less than or equal to $5.

	`slt $3, $5, $4`	`# is $5 < $4?`
	`beq $3, $0, Target`	`# if not, branch to Target`

Part C Write a C code fragment equivalent to the following MIPS code. Register $2 holds the variable "N" and $4 holds the variable "S".

```
.text
 Foo:    addi $4, $0, 0
Loop:    andi $5, $2, 1
         add  $4, $4, $5
         srl  $2, $2, 1
         bne  $2, $0, Loop
```

Equivalent C code:

```
int S = 0; // # of ones
do {
   S += N & 1;
   N >>= 1;
} while (N);
```

Part D Briefly describe what the code fragment in Part C computes.

This routine counts the number of ones in the binary representation of N.