ARM Assembly

```
01:      mystery2
02: CBZ          R0, loc_C672      #compares R0 to 0 and if equal to 0 branch to loc_C672 on line 8
03: LDRB.W       R0, [R0, #0x63]   #Load a byte into R0 from R0+0x63 (99)
04: SUBS         R0, #0            # R0 = R0-0
05: IT NE
06: MOVNE        R0, #1            # R0 = 1 if R0!=0
07: BX           LR                # return
08: loc_C672
09: MOVS         R0, #1            #R0 = 1
10: BX           LR                #return
11:      ;end of function mystery2
```

Mode

This codes mode is thumb mode since it has instructions ranging from 16 bits to 32 bits.

Types

R0 at first is a pointer. After that, R0 is used as a Boolean(0 or 1) for the return value.

Function Prototype

int mystery2(char * arg1)

C Code

```
int mystery2(void * arg1)
{
        if(arg1=='\0')
        {
                return 1;
        }
        printf("%i \n",arg1);
        arg1 = (char *)(arg1+99);
        printf("%s \n",arg1);
        if(*((char *)arg1) != '\0') { return 1;}
        return 0;
}
```

Explanation

This function is given a pointer that points to a struct that has a one byte value at 0x63. If the pointer is NULL, then the value returns true. If it is not NULL, we go to the 100th byte of the struct. If the 100th byte of the struct is 0, the function returns 0. If the last value is not 0, we return 1. The struct could be anything, so I generalized my code so it works with a void pointer. Although, I think this code was intended to check for a null terminating byte at the end of a string. In this case, the string was 100 bytes with the last byte being the null byte.