CS 240
Laboratory 8 Assignment
Disassembly and Reverse Engineering

Analyze the following X86 code for a function ***analyze***, disassembled with **gdb**. Write descriptive comments, and then summarize what you think the function does. Hand in hardcopy before lab.

Assume that the function takes two parameters.

For your analysis, also assume that the function has been invoked with values of 2 and 4.

Dump of assembler code for function **analyze**: **Comments**

```
Dump of assembler code for function analyze:
0x08048414 <+0>:  push   %ebp
0x08048415 <+1>:  mov    %esp,%ebp
0x08048417 <+3>:  sub    $0x28,%esp
0x0804841a <+6>:  movl   $0x1,-0x10(%ebp)
0x08048421 <+13>:jmp    0x804844f <analyze+59>
0x08048423 <+15>:mov    0x8(%ebp),%eax
0x08048426 <+18>:mov    %eax,%edx
0x08048428 <+20>:sar    $0x1f,%edx
0x0804842b <+23>:idivl  -0x10(%ebp)
0x0804842e <+26>:mov    %edx,%eax
0x08048430 <+28>:test   %eax,%eax
0x08048432 <+30>:jne    0x804844b <analyze+55>
0x08048434 <+32>:mov    0xc(%ebp),%eax
0x08048437 <+35>:mov    %eax,%edx
0x08048439 <+37>:sar    $0x1f,%edx
0x0804843c <+40>:idivl  -0x10(%ebp)
0x0804843f <+43>:mov    %edx,%eax
0x08048441 <+45>:test   %eax,%eax
0x08048443 <+47>:jne    0x804844b <analyze+55>
0x08048445 <+49>:mov    -0x10(%ebp),%eax
0x08048448 <+52>:mov    %eax,-0xc(%ebp)
0x0804844b <+55>:addl   $0x1,-0x10(%ebp)
0x0804844f <+59>:mov    -0x10(%ebp),%eax
0x08048452 <+62>:cmp    0x8(%ebp),%eax
0x08048455 <+65>:jg     0x804845f <analyze+75>
0x08048457 <+67>:mov    -0x10(%ebp),%eax
0x0804845a <+70>:cmp    0xc(%ebp),%eax
0x0804845d <+73>:jle    0x8048423 <analyze+15>
0x0804845f <+75>:mov    $0x8048594,%eax
0x08048464 <+80>:mov    -0xc(%ebp),%edx
0x08048467 <+83>:mov    %edx,0x4(%esp)
0x0804846b <+87>:mov    %eax,(%esp)
0x0804846e <+90>:call   0x8048338 <printf@plt>
0x08048473 <+95>:leave
0x08048474 <+96>:ret
```

Description of function: