Reference C++ code

```cpp
int dummy(int x) {
    int ret = x * 19;
    return ret;
}
```

Reference AT&T ASM code

```
_Z5dummyi:
.LFB0:
        .cfi_startproc
        endbr64
        pushq        %rbp
        .cfi_def_cfa_offset 16
        .cfi_offset 6, -16
        movq        %rsp, %rbp
        .cfi_def_cfa_register 6
        movl        %edi, -20(%rbp)
        movl        -20(%rbp), %edx
        movl        %edx, %eax
        sall        $3, %eax
        addl        %edx, %eax
        addl        %eax, %eax
        addl        %edx, %eax
        movl        %eax, -4(%rbp)
        movl        -4(%rbp), %eax
        popq        %rbp
        .cfi_def_cfa 7, 8
        ret
        .cfi_endproc
```

1. **(Max) Given an integer x, can x multiply by 19 be implemented by using shifts and adds only? How?**

   First we define "x multiply by 19" as the equation:

   $$x \cdot 19 = 19x$$

   Notice that 19 can be broken into a sum of 16 and 3 which allows us to rewrite the above to:

   $$19x = (16 + 3)x$$

   $$19x = 16x + 3x$$

   Bit shifting manipulates numbers through powers of 2, we want to break down the expression such that we can express it as the sum of powers of 2. By inspection we can see that 16 is a power of 2. Using the same technique above we can also break down 3.

   $$19x = 2^4x + (2 + 1)x$$

   $$19x = 2^4x + 2^1x + 2^0x$$

   The expression 2^0 is just 1 and x multiplied by 1 is just x, so we end up with the following:

   $$19x = 2^4x + 2^1x + x$$

   Now that we're able to represent the product as a sum of powers of 2, we can then substitute the individual products to their corresponding bit shifts.

   $$19x = (x << 4) + (x << 1) + x$$

2. **(Paco) What does the assembly version do? Does it use the multiply instruction?**

   your mom is a placeholder

3. **(Paco) What happens for the case of $x \cdot 45$?**

   your mom is a placeholder

4. **(Ian) What happens for the case of $x \cdot -2$?**

   your mom is a placeholder

5. **(Max) What happens for the case of $x \cdot 0$?**

   your mom is a placeholder