A function with a prototype

```c
void whatami(int x, int y, int z);
```

is compiled into IA32 assembly code. The body of the code is:

```assembly
x at %ebp+8; y at %ebp+12; z at %ebp+16
```

<table>
<thead>
<tr>
<th>Assembly</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 movl 12(%ebp), %edx</td>
<td>%edx = y;</td>
</tr>
<tr>
<td>2 subl 16(%ebp), %edx</td>
<td>%edx = %edx - z = y - z;</td>
</tr>
<tr>
<td>3 movl %edx, %eax</td>
<td>%eax = y - z;</td>
</tr>
<tr>
<td>4 sall $31, %eax</td>
<td>%eax = %eax &lt;&lt; 31;</td>
</tr>
<tr>
<td>5 sarl $31, %eax</td>
<td>%eax = %eax &gt;&gt; 31;</td>
</tr>
<tr>
<td>6 imull 8(%ebp), %edx</td>
<td>%edx = (y - z) * x;</td>
</tr>
<tr>
<td>7 xorl %edx, %eax</td>
<td>%eax = %edx ^ %eax; (bitwise XOR)</td>
</tr>
</tbody>
</table>

Parameters `x`, `y`, and `z` are stored at memory locations with offsets 8, 12, 16, relative to the address in register `%ebp`. The code stores the return value in register `%eax`.

1. Write C code for `whatami` that will have an effect equivalent to the assembly code.
2. What does this function do?

What does it do?:

In #4: 31 left shifts on a 32-bit register will move the LSB all the way to the MSB, and will fill up all 0s to the right.

In #5: 31 right shifts on the same register will bring the MSB back to its original position as LSB. To the left, the register will be filled up with 0s if the MSB (originally LSB) was 0 (i.e., \(y - z\) = even), all 1s if it was 1 (\(y - z\) = odd).

In #7: The XOR of `%edx` and `%eax` is the XOR of \((y - z) * x\) (in `%edx`) and the result of the 31-shifts-left-31-shifts-right in #4 and #5 (in `%eax`). Thus, the XOR is with either all 0s or all 1s. XOR with 0s preserves the original value; XOR with 1s inverts it (bit-by-bit). So, the final result in `%eax` is

\[
(y - z) * x \text{ if } y - z \text{ is even} \\
\sim((y - z) * x) \text{ if } y - z \text{ is odd}
\]
How I graded: you got

- 4 if you got it right and explained clearly (2 people);
- 3 if you got it more-or-less right (5 people);
- 2 if you showed some reasonably correct direction;
- 1 for showing up, writing something, and turning it in.