|  |
| --- |
| **Skill 6.01 Exercise 1** |
| Consider a computer that uses 6 bits to represent integers: 1 bit for the sign and 5 bits for the actual number. What's the largest positive integer it can represent? |
| If the left-most bit represents the sign, where 0 = positive and 1 = negative, what is 11001 in decimal?  11001 |
| If the left-most bit represents the sign, where 0 = positive and 1 = negative, what is -21 in decimal? |

|  |
| --- |
| **Skill 6.02 Exercise 1** |
| On a computer which uses 6 bits to represent integers (with 1 bit to represent the sign), which of these operations result in overflow? |
| 15+15 |
| 8 \* 4 |
| 3+29 |
| 30+1 |

|  |
| --- |
| **Skill 6.03 Exercise 1** |
| Convert each of the floating point numbers to binary |
| 0.125 |
| 0.03125 |
| 0.675 |
| 0.375 |
| 10.75 |

|  |
| --- |
| **Skill 6.04 Exercise 1** |
| A scientist is running a program to calculate the volume of a cone:  radius ← 17.24  height ← 5.24  volume ← PI \* (radius \* radius) \* (height / 3)  The code relies on the built-in constant PI. After running the code, the variable volume stores 1630.9266447568566.  Their supervisor checks their results by running the same calculation on their own computer. Their program results in a volume of 1630.9266447564448.  The two values are very close, but not quite the same. Why? |
|  |

|  |
| --- |
| **Skill 6.04 Exercise 2** |
| Nuru writes this code to calculate the final cost of an item with a discount applied:  price ← 0.7  discount ← 0.2  final ← price - discount  They're surprised to see that final stores the value 0.49999999999999994 instead of  0.5.  What is the best explanation for that result? |
|  |