Homework 05  
  
1) Suppose I am using Scratch.  
Suppose I have a sprite on the stage.  
How do I move a sprite to a different position on the stage?

***Click and drag the sprite anywhere you want, or you can use the ‘move’ block under the motion scripts.***  
2) Suppose I am using Scratch.  
Suppose I have a sprite on the stage.  
How do I rotate the sprite?

***Use the ‘turn’ block, and set the degrees to the desired rotation.***  
3) Suppose I am using Scratch.  
How do I get blocks inside the script area?

***Click and drag block of choice to script area.***

4)  Suppose I am using Scratch.  
Suppose I have a block in the script area.  
How do I execute/launch/run/invoke a block?

***Click the block you want to execute.***  
5) Some Scratch blocks contain a white oval.  
The white oval contains a number or a string of characters.  
How do I change the contents of the white oval?

***Click the white oval, type in your new value.***

6) Some Scratch blocks contain a white oval with a black arrow.  
The white oval contains a number or a string of characters.  
What is the black arrow?

***Click the black arrow, choose a selection from the popup menu.***

7) Some Scratch blocks contain a white oval with a black arrow.  
The white oval contains a number or a string of characters.  
How do I change the contents of the white oval?

***Click on the white oval and type in a new value.***

8) Suppose I am using Scratch.  
Suppose I have a block inside the script area.  
How do I move the block to a different position in the script area?

***Click and drag the block to wherever you want inside the script area.***  
9) Suppose I am using Scratch.  
Suppose I have a block inside the script area.  
How do I remove the block from the script area?

***Click and drag the block into the center panel, when released it will disappear.***  
10) Suppose I am using Scratch.  
Suppose I have 2 scratch blocks in the script area.  
How do I join the two blocks together?

***Click and drag what is to be the “bottom” block, and move it close to the other block, they will “snap” together.***

11) Suppose I am using Scratch.  
Suppose I have 2 Scratch blocks, in the script area, that are joined together.  
How do I move all the joined blocks to a different position in the script area?

***Click and drag the top most block, any blocks joined to that one will go with it.***

12) Suppose I am using Scratch.  
Suppose I have 2 Scratch blocks, in the script area, that are joined together.  
How do I separate/disjoin/pull-apart the 2 blocks?

***Click and drag the block BELOW where you want it to split. They will detach with the block you clicked on becoming the new “parent” block of that group.***

13) Suppose I am using Scratch.  
Suppose I have 2 Scratch blocks, in the script area, that are joined together.  
How do I delete both blocks?

***Click and drag the “parent” block to the center column, everything joined will go with it, and let go.***

14) Suppose I am using Scratch.  
Suppose I have 2 Scratch blocks, in the script area, that are joined together.  
How do I execute/run/launch/play both blocks?

***Click the “parent” block.***

15) Convert 00100101(base-2) to base-10  
I recommend you show all your work.

**128 64 32 16 8 4 2 1**

**0 0 1 0 0 1 0 1**

***Sum: 32 + 4 + 1 = 37***

16) Convert 165 (base-10) to base-2.  
Please show 8 bits in your answer.  
I recommend you show all your work.

***We start at the largest left-hand bit position that is less than 165:***

***165 – 128 = 37, so in the 128 bit place is 1***

**128 64 32 16 8 4 2 1**

**1 0 0 0 0 0 0 0**

***37 – 32 = 5, so the 32 bit place becomes 1***

**128 64 32 16 8 4 2 1**

**1 0 1 0 0 0 0 0**

***5 – 4 = 1, so the 4 bit place becomes 1***

**128 64 32 16 8 4 2 1**

**1 0 1 0 0 1 0 0**

***1 is left, so the 1 bit place changes to 1***

**128 64 32 16 8 4 2 1**

**1 0 1 0 0 1 0 1**

***Answer: 10100101***  
17) Compute the sum of

**11 1**

**00110110**

**+ 00110100**

**-----------**

**01101010**

18) Compute the difference of

*1 <---Second Carry*

*02202 <---First Carry*

**01001011**

**- 00110110**

**-----------**

**00010101**

19) Write a table that shows the first 16 numbers in base-10, base-2, and base 16.  
Please show 4 bits in your base-2 numbers.

|  |  |  |
| --- | --- | --- |
| base-10 | base-2 | base-16 |
| 0 | 0000 | 0 |
| 1 | 0001 | 1 |
| 2 | 0010 | 2 |
| 3 | 0011 | 3 |
| 4 | 0100 | 4 |
| 5 | 0101 | 5 |
| 6 | 0110 | 6 |
| 7 | 0111 | 7 |
| 8 | 1000 | 8 |
| 9 | 1001 | 9 |
| 10 | 1010 | A |
| 11 | 1011 | B |
| 12 | 1100 | C |
| 13 | 1101 | D |
| 14 | 1110 | E |
| 15 | 1111 | F |

20) Convert 10100101 (base-2) to base-16.

***1010 = 10 = A, 0101 = 5 = 5: ANSWER: A5***

21) Convert A5 (base-16) to base-2.  
Please show 8 bits in your answer;  
I recommend you show all your work.

***A = 10 in base 10, in binary bits 8 and 2 would be ‘on’: 1010***

***For 5, bits 4 and 1 would be ‘on’: 0101***

***Strung together: 10100101***  
22) Convert -37 (base-10) to base-2.  
Please show 8 bits in your answer;  
I recommend you show all your work.

***Ok, 37, bits 32, 4, and 1 would be ‘on’***

**128 64 32 16 8 4 2 1**

**0 0 1 0 0 1 0 1**

**Invert the binary to 11011010**

**11011010+1**

**ANSWER: 11011011**  
23) How are characters stored in a computer?

***As a table of ASCII Characters, denoted by a number.***

**Fun fact: hold ‘alt’ and type the 3 or 4-digit code and you can type ASCII characters: Θ (alt+233), ð (alt+0240), ░ (alt+176), Ɛ (alt+0400)**

24) How are photographs stored in a computer?

***In an array of numbers corresponding to each pixel that makes up the image. The numbers tell the computer what color to use for that pixel.***  
25) How is music or sound stored in a computer?

***Analog sound comes across as fluctuating voltages. You measure this voltage a certain number of times per second, this is called the Sample Rate. We talk about samples per second with the term Herz, 1 herz, or Hz is a “one per second” time interval. Typical sample rates are 44,100Hz (CD Quality), 48,000Hz (Standard sample rate used in DVD, TV, and film).***

***So for a CD, we divide each second of analog sound into 44,100 pieces. The Voltage for each piece is then converted to binary code, and that code is written to the CD.***