Computer Science Illuminated 8th edition Dale Solutions Manual

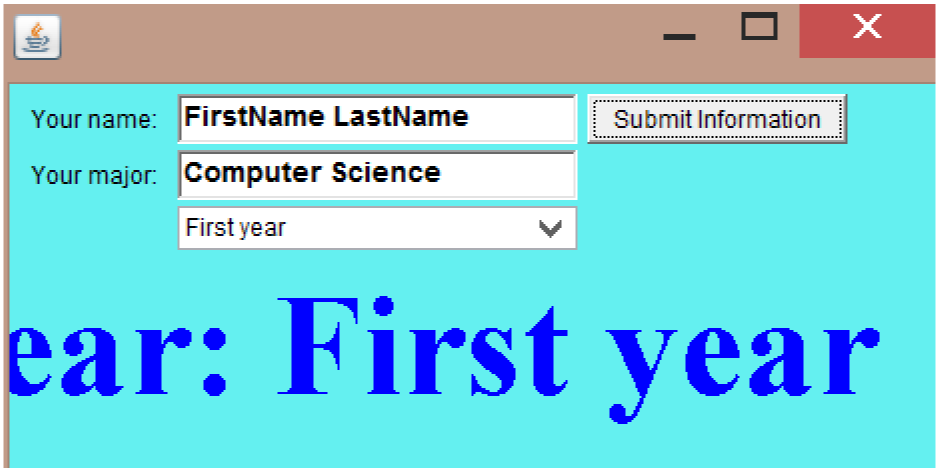
SKU: 9781284275070-SOLUTIONS

Solutions Manual

Computer Science Illuminated, Eighth Edition

Laboratory 1 - Introduction to the Labs

Students will enter basic information in the Intro App and take a screen shot.



Laboratory 2 – Exploring Number Systems

Exercise 1

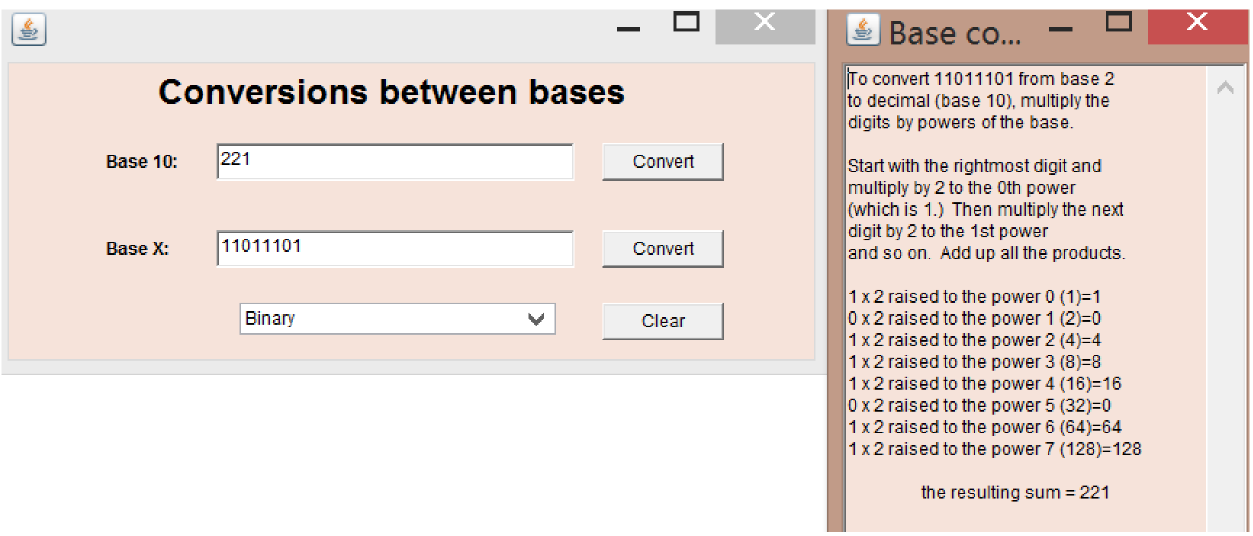
1. Convert the binary number 11011101 to decimal by hand, showing all steps.

   Answer = 221

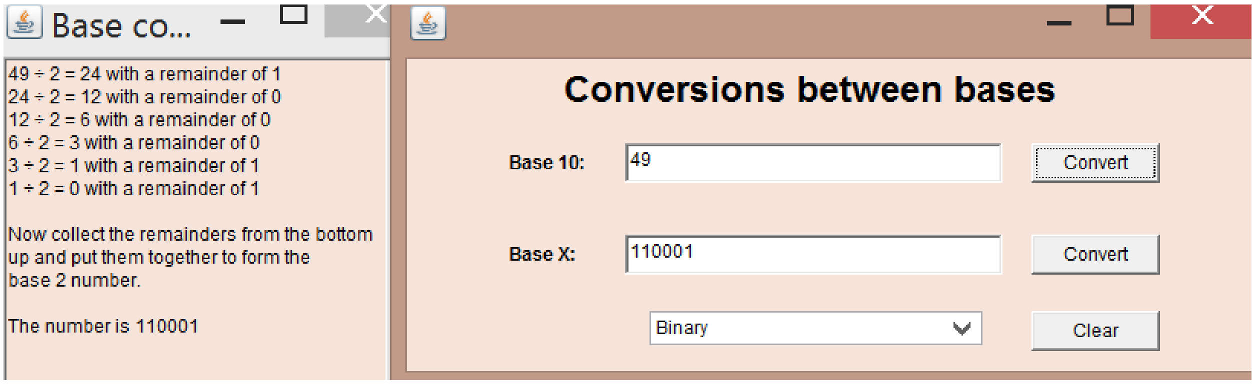
   128 * 1 = 128
   64 * 1 = 64
   32 * 0 = 0
   16 * 1 = 16
   8 * 1 = 8
   4 * 1 = 4
   2 * 1 = 0
   1* 1 = 1

   221

2. Using the “Number systems” app, type 11011101 into the Base X text area. Verify your answer above. Take a screenshot.



3. Example for 49



4. To learn how to do this yourself, convert the decimal number 77 to binary using this method of repeated division, just like the app did.

   77 ÷ 2 = 38 with a remainder of 1

   38 ÷ 2 = 19 with a remainder of 0

   19 ÷ 2 = 9 with a remainder of 1

   9 ÷ 2 = 4 with a remainder of 1

   4 ÷ 2 = 2 with a remainder of 0

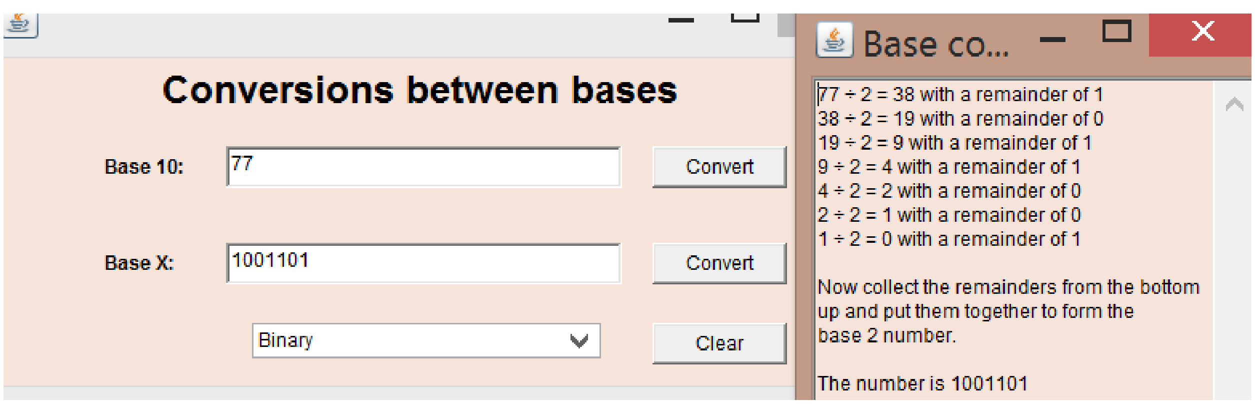
   2 ÷ 2 = 1 with a remainder of 0

   1 ÷ 2 = 0 with a remainder of 1

   Now collect the remainders from the bottom up and put them together to form the base 2 number.

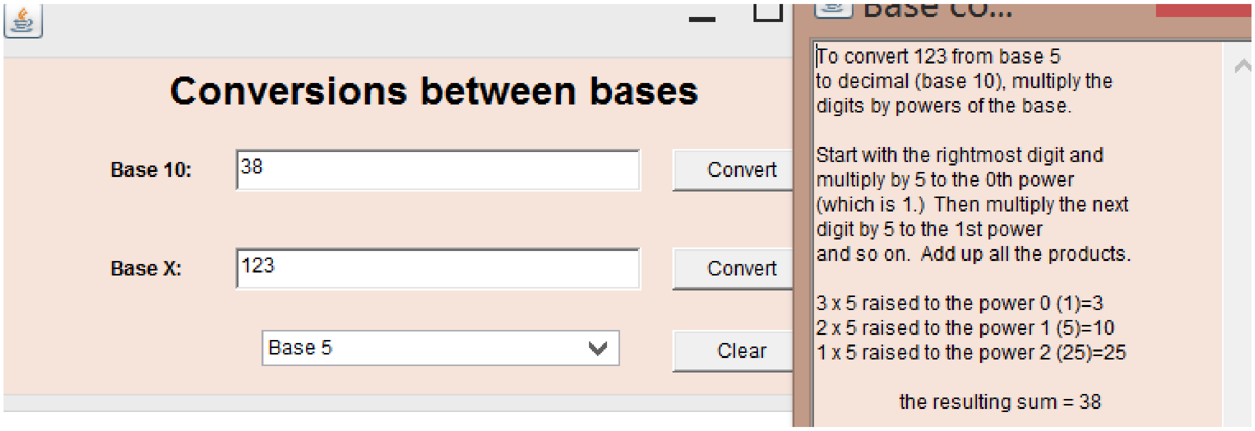
   The number is 1001101

5. Use the “Number systems” app to check your result. Type 77 into the Base 10 text area and press Convert. Take a screenshot.



6. Use the “Number systems” app to convert 123 from base 5 to decimal. Type 123 into the Base X text area, select Base 5 from the choices in the pull-down menu, and click Convert. Does this

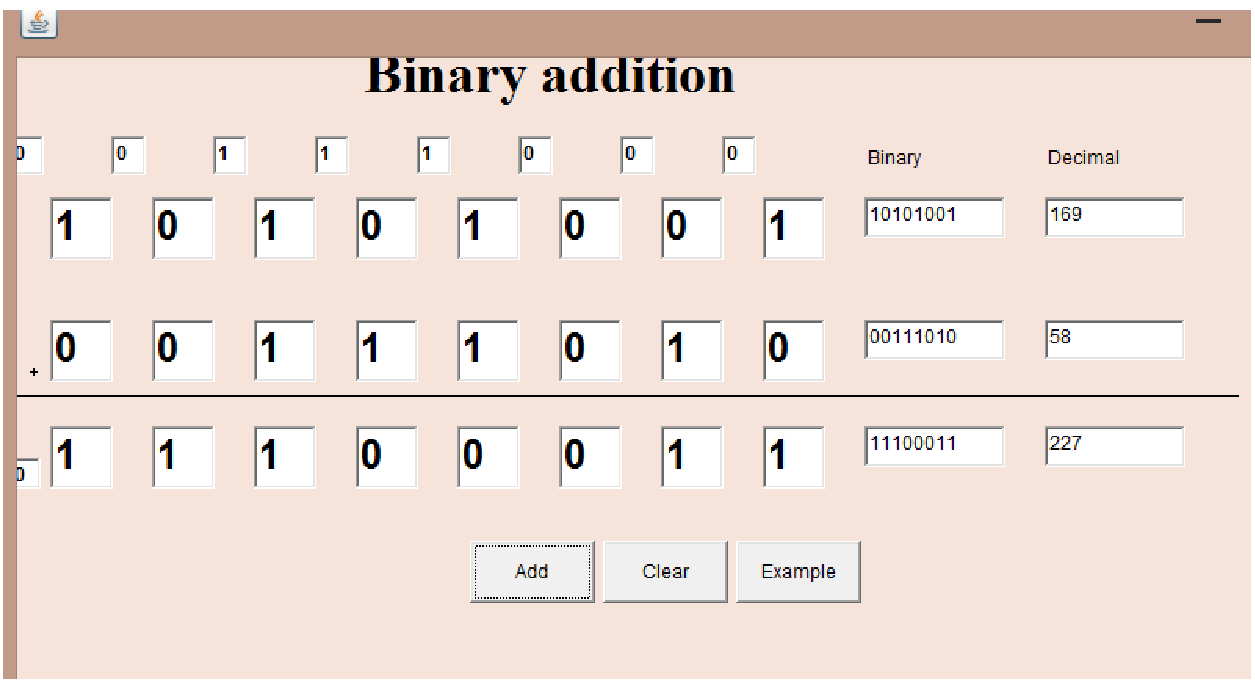
make sense to you? Why? Take a screenshot and write on the paper why 123_5 equals whatever the app shows you.



7. Add the two binary numbers together by hand, showing all carries:

0 1 1 1 0 0 1 0
1 0 1 0 1 0 0 1 1
+ 0 0 0 1 1 1 0 1 0
----
1 1 0 0 0 1 1 0 1

8. Now type the numbers 10101001 and 00111010 into the Binary column of the "Binary addition" app. Click Add and verify your answer with the app's result. Take a screenshot.



9. Finally, what are the limits of the "Binary addition" app? What are the largest numbers it can add? What is the largest result it can produce? Record your answers below, in both binary and decimal values.

Largest numbers it can add __128 + 127 are largest two numbers__

Largest result it can produce ______11111111 255__

Laboratory 3 – Representing Numbers

Exercise 1

1. Start the "Negative binary numbers" app.
2. Type in -1 and press *Return*. Study the result. Find the sign bit. How does the app indicate it?

10000001
The sign bit is the first bit in the series

3. Change the number of bits to 32 in the text field "Number of digits in binary" and convert -1 again. Write down how the two forms of binary -1 are different in 32 bits compared to 8 bits. (Feel free to convert -1 into 8 bits again to see the different forms.)

In binary given 8 digits then 32 digits the structure of the binary number remains the same: sign magnitude form has a 1 in the highest bit (most significant bit) followed by zero's and a 1 at the end. The difference between 8 bits and 32 bits is the number of zeros between the sign bit and the one at the end. One's compliment just increases the number of 1's from 8 to 32.

4. Make a small table as follows (you may delete this sample image from the document and replace it with your own table):
5. Fill in the binary values for these two numbers, changing the number of bits from 8 to 12, then to 16, and then to 32.

	6	-6
8 bits	00000110	10000110
12 bits	000000000110	100000000110
18 bits	000000000000000110	100000000000000110
32 bits	00000000000000000000000000000110	10000000000000000000000000000110

6. Without using the app, guess what -1 and -4 would look like using 64 bits. Write down your guess.

It would be around (with 62 zero's in the middle of both)
1000000000000000000000000000000000000000000000000000000000000001 for -1.
It would be around
1000000000000000000000000000000000000000000000000000000000000110 for -6.

7. Change the number size back to 8 bits. Type in 127 and press *Return*. Describe what you see below.

The sign magnitude form and the 2's complement form both came out the same. 01111111

8. Now type in -127 and convert. Write down the values. What is the difference between +127 and -127 in 2's complement form?

For +127 the 2's complement came out to: 01111111
For -127 the 2's complement came out to: 10000001
For the positive it had much more 1's instead of zeros like the negative did, however it still retains the first digit sign bits of zero for positive and 1 for a negative number. The difference between +127 and -127 is zero in any form.

9. Type 4 into "number of digits in binary." Then convert the number 8. What do you see? Does this seem right?

No, because the number can't fit into 4 bits.

Exercise 2

1. Start the "Real number representations" app
2. Type in a fairly large number, like 8673491502, and press *Return*. Write down the value in scientific notation and in binary.

Scientific notation: 8.673491502000001 x10 ^{9}
Binary: 0 0.1000000100 0100010

3. Experiment with a few other numbers. What generalization can you make about the powers of 10 and 2 that you see? (Hint: Which power is bigger? Is it always bigger by the same amount? Try dividing the bigger by the smaller. Is this ratio always the same or close to the same?)

The POWER of base 2 is LARGER. It is not always larger by the same amount. The ratio of the two powers is not the same. 7/1 = 7, 10/3 = 3.33, 13/3 = 4.33, 17/4 = 4.25

4. How many bits does the binary representation at the bottom of the screen devote to the mantissa (the part after the binary point and separated by a space from the binary power)? What would be the smallest mantissa that could be represented? (Hint: You may need a calculator.)

The smallest mantissa would be 10 digits which is 2^{-10}
Now search the internet for "negative powers of 2" which should return a table. In the table of negative powers of two, when n=10, the number is 0.0009765625 which would be the smallest number given a mantissa of 10.

5. Type a number such as 0.00000000495 and press *Return*. What is the sign of the exponent?