**PROJECT ( Binary to Decimal Conversion Program )**

**Objective** To design / utilize a program to perform binary to decimal number conversions.

***PROJECT DESCRIPTION***

This project has you using and modifying an application that consists of converting a binary number into a decimal number.

***Information About this Project***

Here is an example of converting a binary number into a decimal number.

sample binary number: 01101101

the decimal equivalent of this binary number is computed as follows:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| *power of 2* | 2 7 | 2 6 | 2 5 | 2 4 | 2 3 | 2 2 | 2 1 | 2 0 |
|  |  |  |  |  |  |  |  |  |
| *decimal weight* | 128 | 64 | 32 | 16 | 8 | 4 | 2 | 1 |
|  |  |  |  |  |  |  |  |  |
| *given number* | 0 | 1 | 1 | 0 | 1 | 1 | 0 | 1 |

the sum of the decimal weights is: 0 + 64 + 32 + 0 + 8 + 4 + 0 + 1 = 109

Thus, the decimal equivalent of the binary number 01101101 is 109 .

***Steps to Complete this Project***

**STEP 1**  **Open a Java Compiler on Your Computer**

Launch Eclipse, JCreator, NetBeans or similar Java Compiler.

**STEP 2**  **Execute the Starter Code Program**

Consider the code below in **Figure 1** below. Examine the statements that comprise the program and then copy them into a new Java file.

Execute the program and test it by implementing the binary number that is given in the above example.

**STEP 3**  **Observe the Program Output**

After you run the starter code and test the program with a sample binary number, observe the output, which should appear similar to that given below.

**(Binary to Decimal Converter)**

**Enter an 8 - digit Binary Number**

**Example: 01101101 (positive only)**

**01101101**

**the decimal equivalent of**

**01101101**

**is 109.0**

**PROJECT ( Binary to Decimal Conversion Program )**

**Figure 1 Java Binary to Decimal Conversion Program**

|  |
| --- |
|  |

|  |
| --- |
| **import java.util.Scanner;**  **public class BinaryToDecimal**  **{**  **public static void main(String[] args)**  **{**  **// declare and initialize the program variables**  **Scanner cin = new Scanner(System.*in*);**  **String line = "";**  **char a, b, c, d, e, f, g, h;**  **double decNum = 0.00;**  **String result = "";**  **// display the program menu**  **System.*out*.println("(Binary to Decimal Converter)\n");**  **System.*out*.println("\nEnter an 8 - digit Binary Number\n");**  **System.*out*.println("\nExample: 01101101 (positive only)\n");**    **// input the sample binary number**  **line = cin.nextLine();**  **a = line.charAt(0);**  **b = line.charAt(1);**  **c = line.charAt(2);**  **d = line.charAt(3);**  **e = line.charAt(4);**  **f = line.charAt(5);**  **g = line.charAt(6);**  **h = line.charAt(7);**  **// assign a decimal weight to each digit**  **if(a == '1') { decNum = decNum + 128; }**  **if(b == '1') { decNum = decNum + 64; }**  **if(c == '1') { decNum = decNum + 32; }**  **if(d == '1') { decNum = decNum + 16; }**  **if(e == '1') { decNum = decNum + 8; }**  **if(f == '1') { decNum = decNum + 4; }**  **if(g == '1') { decNum = decNum + 2; }**  **if(h == '1') { decNum = decNum + 1; }**  **// display the program output**  **System.*out*.println("\nthe decimal equivalent of ");**  **result = a + "" + b + "" + c + "" + d + "";**  **result += e + "" + f + "" + g + "" + h;**  **System.*out*.println(result);**  **System.*out*.println("" );**  **System.*out*.println(" is " + decNum);**  **cin.close();**  **}**  **}** |

|  |
| --- |
|  |

**PROJECT ( Binary to Decimal Conversion Program )**

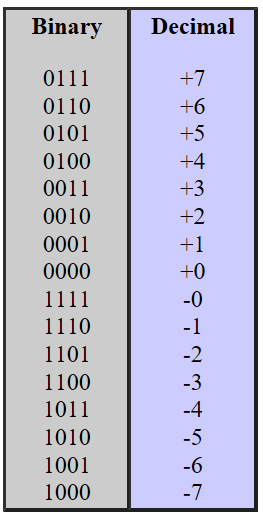
**STEP 4**  **Perform Various Binary to Decimal Conversions**

After you performed a trial run of your program, perform a binary to decimal conversion for five separate binary numbers. Take screen snapshots of your results and paste them into an MS Word lab submittal document.

Use [ Alt ] + [ PrintScreen ] and [ Ctl ] + [ V ] to perform the snapshot and paste routine.

**STEP 5**  **One’s Complement**

Modify your program such that it performs One’s Complements, i.e. each digit of the inputted binary number is flipped 0 to 1 and 1 to 0 . The chart below illustrates the One’s Complement procedure.



Run your modified program and test it with some sample binary numbers. Take screen snapshots of your results and paste them into an MS Word lab submittal document.

**STEP 6**  **Submit Your Assignment for Credit**

Submit both your program source code and your word processing submittal document. Place the files in the appropriate course submittal or drop box.

**Step 7 Questions and Answers Concerning this Computer Laboratory Project**

Open MS Word and, within a new document, place your responses to these questions. Submit your completed MS Word document for credit.

**(1) ( Binary to Decimal )**

In a blank MS Excel worksheet execute the following formula, observe the result and explain the reason why the particular value was returned.

Note: you may have to format the worksheet cell, containing the result,

using a General formatting category to correctly observe the result.

=BIN2DEC(1100101)+BIN2DEC(11010)   
  
**The value returned (127) is the result of converting to decimal form and then adding two \*unsigned\* binary numbers. The unsigned numbers (with zero padding) 01100101 is 101 in decimal form, and 00011010 is 26 in decimal form. If these were signed binary numbers, with a 1 added at the leftmost bit, then the result would be -127.**

**PROJECT ( Binary to Decimal Conversion Program )**

**(2) ( Decimal to Binary )**

To sum two binary numbers, equivalent to 373 10 and 127 10 , acertain student is using the following MS Excel worksheet formula:

=DEC2BIN(373)+DEC2BIN(127)

(a) Using this formula, will the student arrive at the correct binary number result? If not, why?  
  
 **The result (102221212) is incorrect as it is not in binary form.**

**To get the correct result, the student would need to implement**

**Additional logic to do binary arithmetic on the two binary numbers.**

(b) What should be the actual result that the student seeks?  
  
**The actual result should be 101110101 + 1111111 = 0111110100 (500 in decimal)**

**(3) ( Bitwise Shifting )**

Bitwise shift operations move binary number digits to the left or right to form new values.

Here are some examples of such operations:

**[ Programs Code Expressions ]**

**cout << "right shift " << (23 >> 2) << endl;**

**cout << "left shift " << (127 << 3) << endl;**

**[ Program Output ]**



A summary of bitwise shift operations is given below.

*operation notation description*

Right Shift expr1 >> expr2 all bits in expr1 shifted right by expr2 bits

Left Shift expr1 << expr2 all bits in expr1 shifted left by expr2 bits

(a) Determine the result of this bitwise shift operation.

129 >> 3

**16**

(b) Determine the result of this bitwise shift operation.

37 << 4

**592**

**PROJECT ( Binary to Decimal Conversion Program )**

**(4) ( Cryptography: XOR Operations )**

Often used in cryptographic systems, the XOR logical operation has these truth table values.

0 XOR 0 = 0

0 XOR 1 = 1

1 XOR 0 = 1

1 XOR 1 = 0

XOR can be used for bitwise operations.

Visit the following Web site

[**https://www.wolframalpha.com**](https://www.wolframalpha.com)

and execute this command:

213 XOR 140

Observe the result and then explain how the answer was obtained by Wolfram.  
  
 **The bitwise XOR operator (213 ^ 140 in Python) first converts**

**The decimal numbers into binary and pads with zeroes for equal bit depth, and then compares the bit in each position for both the binary values and determines whether they meet the exclusive-or criteria (0 XOR 1 and 1 XOR 0). It then replaces creates a new binary number with the result of the XOR operation in each position, with 1 being a TRUE XOR and 0 being a FALSE XOR.**

**(5) (** **One’s Complement )**

One method for representing signed numbers in the binary number system is One’s Complement. Here, to convert a binary number to One’s Complement we revert each bit of this number, i.e. 0 changes to 1 and 1 changes to 0 . The leftmost bit is 0 for positive numbers and the leftmost bit is 1 for negative numbers. Two representations for zero are used: positive zero ( + 0 ) and negative zero ( − 0 ) .

Decimal − 143

Number of Bits 9

One’s Complement Binary 101110000

For the above scenario and using One’s Complement, determine the complement of 101110000 and then add this result to 101110000 to verify that the sum correctly yields − 0 .   
   
 **The complement of 101110000 (-149) is 010001111 (149) and the sum of these two is: 0**