CSUS

SCHOOL OF ENGINEERING AND COMPUTER SCIENCE

Department of Computer Science

CSC 35

Spring 2021

Dr. Ghansah

**Lab #5: Decimal to Binary Conversion, Input, Output, and ASCII code**

**Purpose:** The purpose of this lab assignment is to learn how to 1) ***input***single ASCII decimal character from the keyboard at the lowest level possible using the Irvine Library call*; 2)* convert a sequence of ASCII decimal digits received from the keyboard to a binary number and use it in a given computation; *3)* convert the binary result of the computation to ASCII decimal digits to be displayed on the screen. The latter was covered in the last lab. This lab is mostly about how to implement algorithm to convert decimal numbers typed from the keyboard as a sequence of ASCII characters, into a binary number. It builds on the previous lab.

**Introduction:** This lab is a modification of the previous lab in the sense that instead of input variables X and Y being hard-coded by assigning them values inside the program, we want to assign values to X and Y on the fly while the program is running. We do so by letting a user type in the values of the variables X and Y from the keyboard. This provides flexibility since we can change X and Y without changing the source code. We will assume the numbers are decimal. Since the number typed from the keyboard consists of a sequence of ASCII decimal digits beginning with the most significant, each typed digit should first be converted to a decimal digit. Moreover, the entire multiple digits typed in has to be converted to a binary before it can be used in a computation. For instance, suppose X=4096. The number 4096 typed on the keyboard will be input as a sequence of ASCII characters ‘4’, ’0’, ‘9’, ‘6’. Eventually we need the binary equivalent 0100 0000 0000 0000 to be placed in a register or in memory. There are a number of algorithms that can be used for this conversion but let us look at one example.

Suppose we want to convert the ASCII string 4096 to a binary number. Let us assume variable X will be used keep track of the binary value. X is initialized to zero. As each ASCII digit is read, we multiply X by 10 and the digit’s binary value is added to X. After all digits have been read, X contains the binary value of the number 4096. Note that the most significant digit is typed first. The following table describes the algorithm.

|  |  |  |
| --- | --- | --- |
| X (before) | New Digit | X (after) |
| 0 \*10 + | 4 | = 4 |
| 4\*10 + | 0 | = 40 |
| 40\*10 + | 9 | = 409 |
| 409\*10 + | 6 | = 4096 |

A pseudo code based on the above table is given below

**Procedure:** Write an assembly program that receives the variables X and Y from the keyboard instead of being hard coded in the program. They are used to compute W as shown below. W is then printed on the screen.

Void main

{

int X, Y, W, loc1, loc2, loc3, sum;

int A=90;

*…*

*Lines of Code to input X from the Keyboard*

*…*

*Lines of Code to input Y from the Keyboard*

*…*

loc1 = Y \* 160 + X \* 2

loc2 = A\* 950

loc3 = loc2 – loc1

Y=3000

Y=Y-1

Sum = loc3/16 +Y + Y/4 + Y/200

W= sum % 7 + 3

Where sum **%** 7 = sum **MOD** 7

…

*Lines of code to output W to the screen*

…

}

**Algorithm for Multi-input Decimal Digits converted to Binary**

Here are some detailed steps to help you input variable X from the keyboard based on the algorithm specified in the table above. For simplicity the following algorithm forces you to type exactly four digits to the screen regardless of the number. Thus, leading zeros are used if the number of decimal digits is less than four. We will learn how to input a variable number of digits without leading zeros in the future.

1. Initialize X=0
2. Set loop counter to 4
3. Multiply X by 10. Make sure you replace X with X\*10.
4. Get Digit from Keyboard (it will be in ASCII)
5. Convert ASCII digit to a number (by subtracting 30h from it)
6. Add Number to X
7. Go to Step 3; You may use the *LOOP instruction* for this.

**Output Session: A typical session on the screen will look like the following**

Input X= 0040

Input Y= 0024

Output W= ?

Input X= 0001

Input Y= 0001

Output W= ?

Input X= 1234

Input Y= 0016

Output W= ?

**HINTS:** Declare variables for strings in the data area. For instance

inputX BYTE “Input X=”, 0

inputY BYTE “Input Y=”, 0

outputW BYTE “Output W=”, 0

newline BYTE 0dh, 0ah, 0

**Note1**: You may not use any library functions other than the ones specified below. For this assignment you will need the following Irvine Library calls.: *ReadChar* for input from keyboard (new for this lab), *WriteChar* for output of single character to screen (used in lab4), and *WriteString* for output of string of characters (eg. Input X=) to the screen. Recall that WriteString was used in Lab2. Check class notes for further details.

**Note2**: Your program should be able to handle up to **4 digit unsigned decimal numbers** for X and Y.

**Testing:** Be sure to test your program and make sure it works before you submit it to your lab instructor on CANVAS as specified below.

**Demonstration**: Demonstrate your program by providing screen shots showing the above session. The lab instructor will assemble and run the documented source code you upload to CANVAS as specified below. The instructor will check whether your program adheres to the session above.

**Submission:** Submit electronic copy of your program to CANVAS including a well *documented program (source code)* and output (screen shots). **Filenames must be according to the format specified in the syllabus**