



Date handed out: 23 March 2023, Thursday

Date submission due: 6 April 2023, Thursday, @23:55, Cyprus Time

Programming Assignment 1: Full Subtractor in Digital Logic

Purpose:

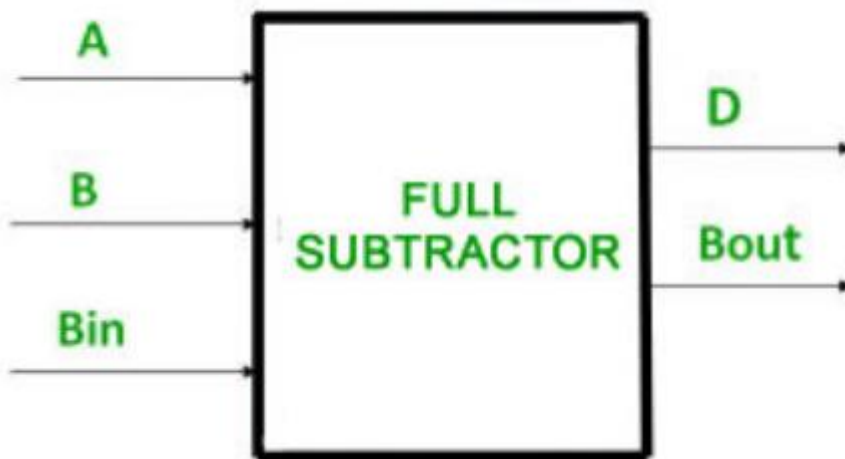
The main purpose of this programming assignment is to revise the topics that we have covered in the first three weeks including fundamentals of C programming, conditional statements and repetitive statements. In this assignment, you will also practice character data types.

Description:

A full subtractor is a combinational circuit that performs subtraction involving three bits, namely A (minuend), B (subtrahend), and Bin (borrow-in). It accepts three inputs: A (minuend), B (subtrahend) and a Bin (borrow bit) and it produces two outputs: D (difference) and Bout (borrow out). In this programming exercise, your task is to write a command-line application that takes the three inputs (A, B and Bin) to a full subtractor, and returns the two outputs (D and Bout).

Programming Requirements:

A full subtractor is designed in such a manner that it can take 3 inputs (A, B and Bin) and produces 2-bit outputs (D, Bout). The figure below shows the block diagram of a full subtractor:



The truth table for the full subtractor is as follows:

INPUT			OUTPUT	
A	B	Bin	D	Bout
0	0	0	0	0
0	0	1	1	1
0	1	0	1	1
0	1	1	0	1
1	0	0	1	0
1	0	1	0	0
1	1	0	0	0
1	1	1	1	1

Figure 1 This figure shows how full subtractor works. Further details available here: <https://www.geeksforgeeks.org/full-subtractor-in-digital-logic/>

When the user first runs the program, you will display the following menu:

```

Welcome to Full Subtractor!
  (1) Compute and Display the outputs
  (2) Quit

```

Depending on the option given by the user, your program will work as follows:

Option 1: When the user chooses this option, your program will first ask the user to enter 3-digit binary inputs (or a number that can be represented in three digits). If more than 3 bits is entered, then your program should print an error message and ask the user to re-enter the input. 3 digits will be used as 3 inputs required by the encoder. First digit is A, the second digit is B and the last digit is Bin.

The inputs can be entered in base 2 or base 16. If the user enters in base 2, then the only allowed digits are 0 and 1 and you will not need to make base conversions. However, if the user enters in base 16, then the only allowed digits are 0-9 and A-F and you will need to convert it to binary. If the user enters a value greater than 7 in base 16¹, then your program will display "Not possible to convert it to 3-digit binary number" and will ask the user to re-enter the value again. The program will work as follows. Further details and cases are given in the sample run below. A sample interaction is as follows:

```

Which base will you use to enter input (base 2 or base 16)? 2
Please enter your input: 101

```

Please note the following cases for this interaction:

If the user enters an invalid input for base 2 or base 16 then your program will display "invalid number in base 2/16" and will ask the user to re-enter the value again.

If the user input is <3 digits in base 2 then your program will display "You entered less than 3 bits! Please try again!" and will ask the user to re-enter the value again.

If the user inputs a value greater than 7 in base 16, then your program should say "Not possible to convert it to 3-digit binary number".

¹ This is because a value greater than 7 requires more than 3 digits to be presented in binary.

After the program receives the input successfully, it needs to calculate and return D and Bout values. The value of D and Bout will be calculated as follows:

$$D = (A \text{ XOR } B) \text{ XOR } \text{Bin}$$

$$\text{Bout} = (\text{NOT}(A) \text{ AND } \text{Bin}) \text{ OR } (\text{NOT}(A) \text{ AND } B) \text{ OR } (B \text{ AND } \text{Bin})$$

Please note that when all the inputs are zero, an output with all 0's is generated.

For your reference, the truth table for bitwise OR, AND, XOR and NOT are as follows:

X(where X is a single bit)	Y(where Y is a single bit)	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

X(where X is a single bit)	Y(where Y is a single bit)	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X(where X is a single bit)	Y(where Y is a single bit)	X XOR Y
0	0	0
0	1	1
1	0	1
1	1	0

X(where X is a single bit)	NOT X
0	1
1	0

Please note that you cannot use the bitwise and/or/xor operations provided in C. You can use logical AND, logical OR operators.

If the user selects **option 2 (Quit)**, then your program will stop.

Sample Run-1:

```
Welcome to Full Subtractor!
```

```
(1) Compute and Display the outputs
```

```
(2) Quit
```

```
You choose: 1
```

```
You have chosen option 1
```

```
Which base will you use to enter input (base 2 or base 16)? 2
```

```
Please enter your input: 001
```

```
D is 1 Bout is 1
```

```
Welcome to Full Subtractor!
```

```
(1) Compute and Display the outputs
```

```
(2) Quit
```

You choose: 1
You have chosen option 1

Which base will you use to enter input (base 2 or base 16)? 16
Please enter your input: F
Not possible to convert it to 3-digit binary number. Please try again!
Please enter your input: 5
D is 0 Bout is 0

Welcome to Full Subtractor!

(1) Compute and Display the outputs
(2) Quit
You choose: 1
You have chosen option 1

Which base will you use to enter input (base 2 or base 16)? 2
Please enter your input: 111
D is 1 Bout is 1

Welcome to Full Subtractor!

(1) Compute and Display the outputs
(2) Quit
You choose: 1
You have chosen option 1

Which base will you use to enter input (base 2 or base 16)? 2
Please enter your input: 22
Invalid number in base 2! Please try again!
Please enter your input: 1110001
You entered more than 3 bits! Please try again!
Please enter your input: E
Invalid number in base 2! Please try again!
Please enter your input: 011
D is 0 Bout is 1

Welcome to Full Subtractor!

(1) Compute and Display the outputs
(2) Quit
You choose: 1
You have chosen option 1

Which base will you use to enter input (base 2 or base 16)? 2
Please enter your input: 01
You entered less than 3 bits! Please try again!
Please enter your input: 000
D is 0 Bout is 0

Welcome to Full Subtractor!

(1) Compute and Display the outputs
(2) Quit
You choose: 1
You have chosen option 1

Which base will you use to enter input (base 2 or base 16)? 16
Please enter your input: K
Invalid number in base 16! Please try again!
Please enter your input: 7
D is 1 Bout is 1

Welcome to Full Subtractor!

```

(1) Compute and Display the outputs
(2) Quit
You choose: 1
You have chosen option 1

Which base will you use to enter input (base 2 or base 16)? 2
Please enter your input: 451
Invalid number in base 2! Please try again!
Which base will you use to enter input (base 2 or base 16)? 2
Please enter your input: 000
D is 0 Bout is 0

Welcome to Full Subtractor!

(1) Compute and Display the outputs
(2) Quit
You choose: 2
You have chosen option 2
BYEE!!

```

Grading Schema:

If your code does NOT compile, you will automatically get zero. If your code compiles, you will then be graded based the following scheme:

Grading Point	Mark (100)
The menu(keeping the users in a loop until exit is chosen)	10
Reading input	10
Converting to base 2	25
Displaying error messages	20
Finding the output	25
Code quality (e.g., formatting, commenting, naming variables, clean use of C constructs such as formulation of selection statements and loops, etc) ²	10

If you do not obey the rules then you will automatically get 0.

Rules:

Please make sure that you follow the restrictions for the assignment as follows.

- **Strictly obey the input output format. Do not print extra things.**
- **You are not allowed to use global variables and goto statements.**
- **You are not allowed to use data structures such as arrays/strings to store values as we have not covered them in the class yet.**
- **You are not allowed to define your own functions as we have not covered them in the class yet.**
- **You cannot use the bitwise and/or/xor operations provided in C.**
- **Add your name/surname and ID at the top of your code as comments and name your source file "Name-Surname-StudentID.c".**
- **Submit your solution as C and PDF to odtuclass. Do not compress it (zip, rar, ...).**

² See guidelines given here: https://www.gnu.org/prep/standards/html_node/Writing-C.html