Date handed out: **23 July 2024, Tuesday**
Date submission due: **31 July 2024, Wednesday, @23:55**

Programming Project 1: One to Four Demultiplexer

Purpose:

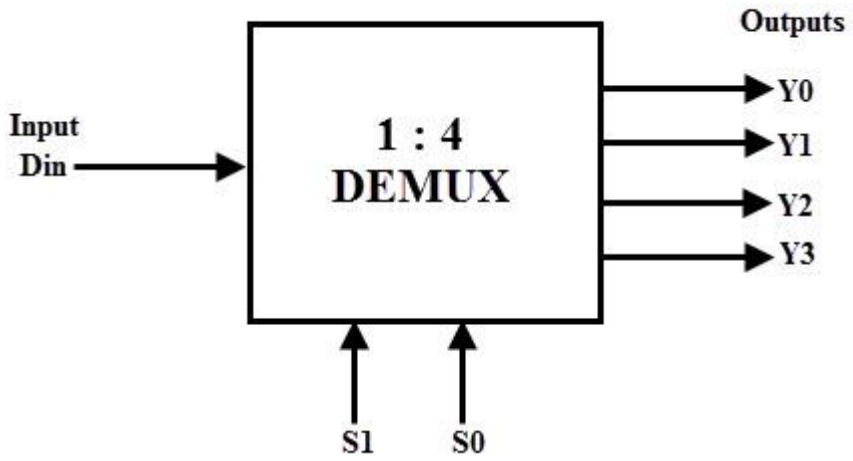
The main purpose of this programming assignment is to revise the topics that we have covered in the first three weeks including fundamentals of C programming, conditional statements and repetitive statements. In this assignment, you will also practice character data types.

Description:

A 1-to-4 demultiplexer has a single input (D), two selection lines (S1 and S0) and four outputs (Y0 to Y3). The input data goes to any one of the four outputs at a given time for a particular combination of select lines.In this programming exercise, your task is to write a C program that takes the input and two selection lines to a demultiplexer, and returns the four output generated by a demultiplexer.

Programming Requirements:

A 1 to 4 demultiplexer is designed in such a manner that it can take 1 input (D) and 2 selection lines (S0 and S1),and produce 4-bit outputs (Y3,Y2,Y1,Y0).
The figure below shows the block diagram of a octal to binary encoder:



The truth table for the encoder is:

S1	S0	D	Y3	Y2	Y1	Y0
0	0	0	0	0	0	**0**
0	0	1	0	0	0	**1**
0	1	0	0	0	**0**	0
0	1	1	0	0	**1**	0
1	0	0	0	**0**	0	0
1	0	1	0	**1**	0	0
1	1	0	**0**	0	0	0
1	1	1	**1**	0	0	0

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

Welcome to 1 to 4 Demultiplexer!
a) Compute and Display the outputs
b) Quit

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

Option 'a': When the user chooses this option, your program will first ask you to enter 3-digit binary inputs. 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 demultiplexer. First digit is S1, the second digit is S0 and the last digit is D.

The inputs can be entered in base 2 or base 10. If the user enters in base 2, then you will not need to make base conversions. However, if the user enters in base 10, then you will need to convert it to binary. If the user enters a value >7 in base 10, 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.

Which base will you use to enter input (base 2 or base 10)? 2
Please enter your input: 010

From the above table, the output Y3 becomes 1 if all of the digits S1and S0 and D are one. Thus, we can write its expression as

Y0= (NOT S1) AND (NOT S0) AND D

Y1= (NOT S1) AND S0 AND D

Y2= S1 AND (NOT S0) AND D

Y3= S1 AND S0 AND D

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

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)	NOT X
0	1
1	0

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

If the user enters an invalid input for base 2 or base 10 then your program will display "invalid number in base 2/10" 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 input is >3 digits in base 2 then your program will display "You entered more than 3 bits! Please try again!" and will ask the user to re-enter the value again.

Please note that if the user selects an option other than a or b, then your code should give an error message.

Please assume that user inputs only positives,not negatives for the inputs.

Sample Run:

```
Welcome to 1 TO 4 DEMULTIPLEXER!

   a) Compute and Display the outputs
   b) Quit
You choose: a
You have chosen option a

Which base will you use to enter input (base 2 or base 10)? 2
Please enter your input: 010
Y3 is 0 Y2 is 0 Y1 is 0 Y0 is 0

Welcome to 1 TO 4 DEMULTIPLEXER!

   a) Compute and Display the outputs
   b) Quit
You choose: d
Invalid option ! Chooese either a or b!


You choose: a
You have chosen option a

Which base will you use to enter input (base 2 or base 10)? 10
Please enter your input: 300
Decimal 300 cannot be represented with 3 bits. Please try again!
Please enter your input: 5
Y3 is 0 Y2 is 1 Y1 is 0 Y0 is 0


Welcome to 1 TO 4 DEMULTIPLEXER!

   a) Compute and Display the outputs
   b) Quit
You choose: a
You have chosen option a
```

```
Which base will you use to enter input (base 2 or base 10)? 10
Please enter your input: 0
Y3 is 0  Y2 is 0 Y1 is 0 Y0 is 0

Welcome to 1 TO 4 DEMULTIPLEXER!

  a) Compute and Display the outputs
  b) Quit
You choose: a
You have chosen option a

Which base will you use to enter input (base 2 or base 10)? 2
Please enter your input: 22
22 is not a valid number in base 2! Please try again!
Please enter your input: 11100
You entered more than 5 bits! Please try again!
Please enter your input: 011
Y3 is 0 Y2 is 0 Y1 is 1 Y0 is 0

Welcome to 1 TO 4 DEMULTIPLEXER!

  a) Compute and Display the outputs
  b) Quit
You choose: a
You have chosen option a

Which base will you use to enter input (base 2 or base 10)? 2
Please enter your input: 01
You entered less than 3 bits! Please try again!
Please enter your input: 111
Y3 is 1 Y2 is 0 Y1 is 0 Y0 is 0


Welcome to 1 TO 4 DEMULTIPLEXER!


  a) Compute and Display the outputs
  b) Quit
You choose: b
You have chosen option b
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:

Your program will be graded as follows:

Grading Point	**Mark (100)**
The menu(keeping the users in a loop until quit 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)[1]	10

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

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 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 are not allowed to use bitwise operators.**
- **Add your name/surname and ID at the top of your code as comments and name your source file "main.c".**
- **Submit your solution as a C file to odtuclass. Do not compress it (zip, rar, …).**

If you fail to obey any of the above rules, you will automatically get zero.