MIDDLE EAST TECHNICAL UNIVERSITY, NORTHERN CYPRUS CAMPUS

CNG140 C Programming - Programming Assignment 1

Date handed out: Saturday 11 March 2017

Date submission due: Saturday 25 March 2017

Programming Assignment 1: 4 to 1 Multiplexer

Purpose:

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

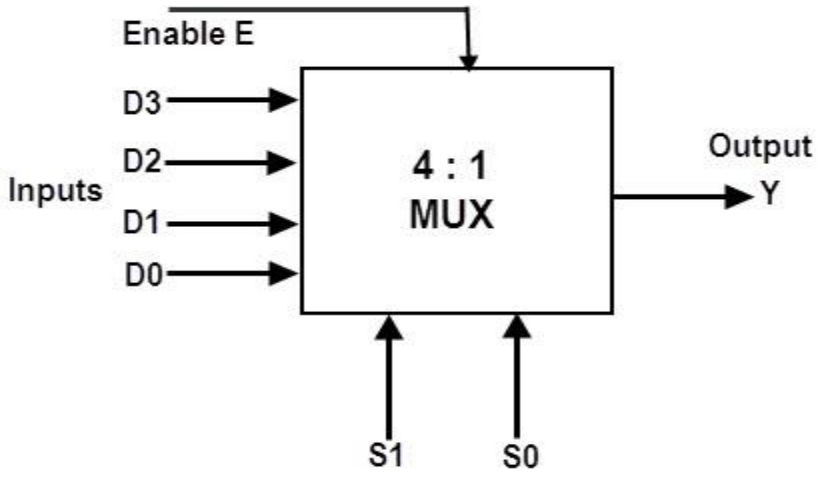
Description:

A **multiplexer** (or **mux**) is a device that selects one of several analog or digital input signals and forwards the selected input into a single line. In this assignment, you will write a small C program for a 4 to 1 multiplexer.

Programming Requirements:

A 4-to-1 multiplexer consists four data input lines as D0 to D3, two select lines as S0 and S1 and a single output line Y. The select lines S1 and S2 select one of the four input lines to connect the output line. The particular input combination on select lines selects one of input (D0 through D3) to the output.

The figure below shows the block diagram of a 4-to-1 multiplexer in which the multiplexer decodes the input through select line.



The truth table of a 4-to-1 multiplexer is shown below in which four input combinations 00, 10, 01 and 11 on the select lines respectively switches the inputs D0, D2, D1 and D3 to the output. That means when S1=0 and S0 =0, the output at Y is D0, similarly Y is D1 if the select inputs S1=0 and S0= 1 and so on.

Select Data Inputs		Output
S_1	S_0	Y
0	0	D_0
0	1	D_1
1	0	D_2
1	1	D_3

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

4 to 1 Multiplexer!

(1) Compute and display the output

(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 you to enter a **2-digit** binary number as the select lines. The user should enter exactly a 2-digit binary number, otherwise an error message will be displayed and the program will ask the user to reenter the select lines.

Please enter select lines: 10

Then, your program will ask which base you will use to enter data lines as follows:

Which base will you use to enter data lines (base 16/10/2)? 2

Please enter the data lines to decode: 0110

The data lines can be entered in different bases, in base 2, base 10 or base 16. If the user is entering the text in base 2, then you will **not** need to make base conversions. However, if the user is entering in base 10 or base 16, then you will need to convert it to binary. Assuming that the user is entering more than 4 bits for data lines, then you program should print an error

message and ask the user to enter the values again. If the data lines are entered in base 16 or 10, your program will convert it to 4- digit binary number. Assume that 4-digit binary number is 0100 then D0=0, D1=0, D2=1 and D3=0. If a value > 15 is entered, then your program will print a message "not possible to convert it to 4-digit binary number" and will ask the user to enter the value again.

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

Full Sample Run:

```
Welcome to 4 to 1 multiplexer!
(1) Compute and Display the Output
(2) Quit
You choose: 1
You have chosen option 1
Please enter select lines: 10
Which base will you use to enter data lines (base 16/10/2)? 2
Please enter the data lines to decode: 0110
Output is 1

Welcome to 4 to 1 multiplexer!
(1) Compute and Display the Output
(2) Quit
You choose: 1
You have chosen option 1
Please enter select lines: 101
It is not a 2-digit binary number for the select lines. Please try again!
Please enter select lines: 01
Which base will you use to enter data lines (base 16/10/2)? 16
Please enter the data lines to decode: A
Output is 1

Welcome to 4 to 1 multiplexer!
(1) Compute and Display the Output
(2) Quit
You choose: 1
You have chosen option 1
Please enter select lines: 00
Which base will you use to enter data lines (base 16/10/2)? 10
Please enter the data lines to decode: 20
It is not possible to convert decimal 20 to 4-digit binary number. Please try
again!
Please enter the data lines to decode: 12
Output is 0

Welcome to 4 to 1 multiplexer!
(1) Compute and Display the Output
(2) Quit
You choose: 2
You have chosen option 2
Byee!!
```

Grading Schema:

Your program will be graded as follows:

Grading Point	**Mark (100)**
The menu (keeping the users in a loop until exit is chosen)	10
Reading text in binary	10
Converting to correct base	25
Displaying error messages	10
Finding the Output of the Mux	25
Code quality (e.g., variable names, formulation of selection statements and loops, etc)	20

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 data structures such as arrays to store values for the conversion operation.**
- **You are not allowed to define your own functions.**
- Name your source file "CNG140-P1.c"
- Upload only source file. Do not compress it (zip, rar, …)