DIU

Department of Computer Science and Engineering

**CSE 123**

**LAB 2**

**Objective**

In the last two labs, we learnt about **the basic building blocks** of all C programs – **literals, data types, variables and operators**. We then wrote a few simple programs of our own, and tested that they are functioning properly. All the programs that we wrote up to last week had something in common – they had a **completely sequential structure**. This means that these programs followed all steps, one after another, from start to end. No matter how many times we run the programs, any particular step would not be skipped or repeated more than once. Today, we will introduce you to programs that have a more flexible structure. In particular, we will see how to write programs where some steps are only executed based on conditions during the program runtime.

*Before we move on, please make sure you have done all the required readings*

**Exercise Set A**

**Task 1:**

Write a C program that reads two integers from the user and prints “first” if the first number is greater than the second number.

**Task 2:**

Write a C program that reads two integers from the user. Your program should then print “first is greater” if the first number is greater, and “first is not greater” otherwise.

**Task 3:**

Write a C program that reads two integers from the user. Your program should then print “first is greater” if the first number is greater, “second is greater” if the second number is greater, and “the numbers are equal” otherwise.

**Task 4:**

Write a C program that reads two integers, subtracts the smaller number from the larger one, and prints the result.

**Task 5:**

Repeat all the four tasks above for two float numbers.

**Task 6:**

Write a C program that reads an integer, and prints “The number is even” or “The number is odd”, depending on whether the number is even or odd. (Hint: use the modulus operator)

**Task 7:**

Write a C program that reads an integer, and if the number is even and greater than 10, prints “An even number greater than 10”. If the number is even but lesser than 10, print “An even number not greater than 10”. If the number is greater than 10 but odd, print “An odd number greater than 10”. If the number is odd and also less than 10, print “An odd number less than 10”.

**Task 8.1:**

Write a C program that reads an integer, and prints the integer if it is a multiple of **either 2 or 5**.

For example, 2, 4, 5, 6, 8, 10, 12, 14, 15, 16, 18, 20, 22 …

**Task 8.2:**

Write a C program that reads an integer, and prints the integer if it is a multiple of either 2 or 5 **but not both.**

For example, 2, 4, 5, 6, 8, 12, 14, 15, 16, 18, 22 …

**Task 8.3:**

Write a C program that reads an integer, and prints the integer if it is a multiple of 2 **and** 5.

For example, 10, 20, 30, 40, 50 …

**Task 8.4:**

Write a C program that reads an integer, and prints the integer if it is a multiple of **NEITHER 2**

**NOR 5.**

For example, 1, 3, 7, 9, 11, 13, 17, 19, 21, 23, 27, 29, 31, 33, 37, 39 …

**Task 8.5:**

Write a C program that reads an integer, and prints the integer if it is **NOT** a multiple of **2** **OR**

**NOT** a multiple of **5**.

**Task 9:**

Write a C program that reads a student’s mark for a single subject, and prints out “Pass” if the student got more than 50, and “You shall not pass” otherwise.

**Task 10:**

Write a C program that reads a student’s mark for a single subject, and prints out the corresponding grade for that mark. The mark ranges and corresponding grades are shown in the table below.

|  |  |
| --- | --- |
| **Marks** | **Grade** |
| 90 and above | A |
| 80-89 | B |
| 70-79 | C |
| 60-69 | D |
| 50-59 | E |
| Below 50 | F |