National University of Computer and Emerging Sciences

Lab Manual # 2

Programming Fundamentals

(Section BCS-1H1&1H2)

Course Instructor	Ms. Arooj Khalil
Lab Instructors	Nimra Abbas Saleha Batool
Section	BCS-1H1&1H2
Semester	Fall 2022

Department of Computer Science
FAST-NU, Lahore, Pakistan

Instruction/Notes: Cheating during the lab will result in negative marks

Topics Covered: Computational problem solving and writing C++ programming using variables, expressions, and conditional execution.

Objectives:

The major focus of this lab is on Converting Algorithms into working C++ code and writing Algorithms using ALGORITHM 2.0.

To write the working C++ code, you are required to use Visual Studio or the online IDE available at https://www.onlinegdb.com/

Question#1 (20 marks)

In the Gregorian calendar, every year that is exactly divisible by four is a leap year, except for years that are exactly divisible by 100 (i.e., end-of-century years). These years are leap years if they are exactly divisible by 400. For example, the years 1700, 1800, and 1900 are not leap years, but the years 1600 and 2000 are. This logic can be easily converted into an Algorithm program that can determine if a year is a leap year or otherwise.

The following Algorithm has been written for this purpose and you are required to write a program using C++ to do the same.

```
1  CR Year
2  IN Year
3  IF ( Year MOD 400 == 0) THEN
       OUT Year, “IS A LEAP YEAR”
   ELSE
        IF ( Year MOD 100 == 0) THEN
           OUT Year, “IS NOT A LEAP YEAR”
   ELSE
       IF ( Year MOD 4 == 0) THEN
          OUT Year, “IS A LEAP YEAR”
   ELSE
         OUT Year, “IS NOT A LEAP YEAR”
         END IF
     END IF
   END IF
4  OUT “THE END”
```

Question#2 (20 marks)

Digital computers represent the different characters of the English language and digits from 0 to 9 using numeric codes (i.e., numbers). One possible way to represent such information is using ASCII (American Standard Codes for Information Interchange). The following table shows the range of ASCII values for various characters.

Characters	ASCII Values
A – Z	65 – 90
a – z	97 – 122
0 – 9	48 – 57
special symbols	0 - 47, 58 - 64, 91 - 96, 123 – 127

In summary, the codes 65 to 90 are used for capital letters, codes 97 to 122 are used to represent small letters, codes 48 to 57 are used to represent numeric digits and all other codes below 128 are used to encode special characters. The following pseudo-code can be used to determine the type (i.e., capital, small, digit, or special) of a character coded by a given number. Convert this pseudo code into a working C++ program.

```
1       CR Code
2       IN Code
3       IF ( Code > 64 AND code < 91) THEN
                OUT Code , “Represents an upper case English alphabet”
        ELSE
                IF (Code > 96 AND code < 123) THEN
                        OUT Code , “Represents a lower case English alphabet”
                ELSE
                        IF (Code > 47 AND code < 58) THEN
                                OUT Code, “Represents a digit”
                        ELSE
                                IF ( Code < 128) THEN
                                        OUT Code, “Represents a special character”
                                ELSE
                                        OUT Code, “DOES NOT REPRESENT AN ASCII Character”
                                END IF
                        END IF
                END IF
        END IF
4        OUT “THE END”
```

Question#3 (20 marks)

It is a known fact that the sum of angles of a triangle is 180° and for any three values that add to 180 one can construct a triangle with angles equal to the given values.

The Algorithm given below can be used to determine if a triangle can be created with angles equal to the values specified by the user. Use it to create a C++ program that asks the user to specify three positive angles and then displays a message stating if a triangle can be created using these angles or not.

```
1       CR A, B, C
2       IN A, B, C
3       IF (A <= 0 OR B<=0 OR + C<=0) THEN
                OUT “ALL ANGLES MUST BE POSITIVE”
        ELSE
                IF (A + B + C == 180) THEN
                        OUT “A Triangle can be created using these angles”
                ELSE
                        OUT “A Triangle cannot be created using these angles”
                END IF
        END IF
4       OUT “THE END”
```

Question#4 (20 marks)

Write an Algorithm using ALGORITHM 2.0 and the corresponding C++ program that inputs a six-digit positive integer and prints the sum, and product of its six digits. Make sure that the number has to be a six-digit number i.e. the numbers are in the range (100000 to 999999).

Sample Input: 153426
Sample output: SUM: 21, PRODUCT: 720

Sample Input: 15342
Sample output: NOT A SIX DIGIT NUMBER: TOO SMALL

Sample Input: 1534269
Sample output: NOT A SIX DIGIT NUMBER: TOO LARGE

HINTS: DIV (C++ integer division i.e., /) and MOD (C++ integer mod i.e., %) operations might be useful for this purpose