Programming Fundamentals – Spring 2021

(BS-IT-F20 Morning & Afternoon)

Assignment # 2

Assigned on: Monday, July 12, 2021
Submission Deadline: Monday, July 30, 2021 (till 11:49 PM)

Instructions

- This is an individual assignment. You are NOT allowed to work/submit in the form of group. Absolutely NO collaboration is allowed. Any traces of plagiarism/cheating would result in an "F" grade in this course.
- Do **NOT** copy even a single line of code from any other person or book or Internet or any other source.
- Read Submission Procedure at the end of File Carefully.
- Late submissions will NOT be accepted, whatever your excuse may be.

Task 1: Multiplication without *

Write a program which ask the user to enter two positive integers. Then you have to calculate their product without using symbol (*) and print the result on screen.

Input validation: Your program has to ask the user again and again to enter positive integers unless user enters valid integers.

Hint: Think of repetitive addition.

Sample Run:

```
Enter first number: -1
Invalid Input! Enter first number
again: 12
Enter second number: 3

Product is:  36
```

Task 2: Division without /

Write a program which ask the user to enter a positive dividend. After that the program ask the use to enter a positive divisor and you have to validate whether the divisor is less that or equal to the dividend or not. Then you have to calculate their remainder without modulus symbol (%) and quotient without divide symbol (/) and print the result on screen.

Input Validation: Your program has to ask the user again and again to enter positive dividend and divisor unless user enters valid integers. And also validate that the divisor must be less than or equal to the dividend.

Hint: Think of repetitive subtraction.

Sample Run:

```
Enter Dividend: -3
Invalid Input! Enter Dividend
Again: 14
Enter Divisor: 18
Invalid Input! Enter Divisor
Again: 5

Quotient: 2
Remainder: 4
```

Task 3: Addition by Hand

What is addition by Hand: All of you have used this method of adding two numbers in your childhood. In this method we start from right side of both numbers and add their last digit and then add the carry of their sum to the second last digits sum and this process goes on unless both numbers end.

Write a program which ask the user to enter two positive numbers. Then you have to calculate their sum by using the above explained method and print the result on screen.

Input validation: Your program has to ask the user again and again to enter positive integers unless user enters valid integers.

Note: Both numbers may not be of same length.

Sample Run:

```
Enter First Number: -7
Invalid Input! Enter Number
Again: 14567
Enter Second Number: 1234
  14567
+  1234
----------
  15801
```

Post Assignment Work: You can also Perform the multiplication by Hand by using the same procedure.

Task 4: Binary Addition

Write a program which ask the user to enter two binary numbers. Then you have to calculate their sum and print the result on screen.

Assumption: The user always enters valid binary number (combination of only 0's and 1's)

Note: Both binary numbers may not be of same length.

Hint: Binary Addition can be done same as we did the above task (Sum by Hand) with some rules which are explained below.

1) If digit1 is 0 and digit2 is 0 and the carry of previous addition is 0 then the sum is 0 and carry is also 0.

2) If digit1 is 1 and digit2 is 0 and the carry of previous addition is 0 then the sum is 1 and carry is 0.

3) If digit1 is 1 and digit2 is 1 and the carry of previous addition is 0 then the sum is 0 and the carry is 1.

4) If digit1 is 1 and digit2 is 1 and the carry of previous addition is 1 then the sum is 1 and the carry is also 1.

5) If digit1 is 1 and digit2 is 0 and the carry of previous addition is 1 then the sum is 0 and carry is 1.

6) If digit1 is 0 and digit2 is 1 and the carry of previous addition is 1 then the sum is 0 and carry is 1.

Sample Run:

```
Enter First Binary Number:
1110011
Enter Second Number: 111010

   1110011
+   111010
--------------
  10101101
```

Task 5: Prime Number

Write a program that ask the user to enter a positive integer and prints whether the integer is prime or not a prime.

Note: An integer is said to be prime if it has exactly 2 divisors. For example, 5 is a prime number because it has only two divisors 1 and 5.

Hint: Use a loop.

Sample Run:

```
Enter a Positive Integer: -1
Invalid Input! Enter Positive
Integer Again: 17

17 is a Prime Number.
```

Task 6: Twin Primes

Write a program which ask the user to enter a positive integer n. Then you have to print all the twin primes from 2 to till the n(inclusive).

Note: Prime numbers which have a difference of 2 are called twin primes. For example, (3,5) is a twin prime.

Hint: You can use the above task (Prime Number) to do this task. Use a nested Loop.

Sample Run:

```
Enter a Positive Integer: 19
Twin Primes between 2 and 19 are:
(3,5)
(5,7)
(11,13)
(17,19)
```

Task 7: LCM (least Common Multiple)

Write a program which ask the user to enter two positive integers. Then you have to calculate their LCM and print the result on screen.

Note: LCM is the least number which is exactly divisible by both numbers.

Hint: Use a loop.

Sample Run:

```
Enter first number: -1
Invalid Input! Enter first number
again: 8
Enter second number: 12

LCM is: 24
```

Task 8: Digit, Alphabet, Others

Write a program which ask the user to enter a character. Then your display the message on screen whether it is a digit, small alphabet, capital. If the character is not digit or alphabet then you simple have to print It is some other Character.

Hint: Think of **ASCII** Values of Characters.

Sample Run:

```
Enter a Character: 8
It is a Digit!
Do you want to enter again(Y/N): Y

Enter a Character: T
It is a Capital Alphabet!
Do you want to enter again(Y/N): Y

Enter a Character: a
It is a small Alphabet!
Do you want to enter again(Y/N): Y

Enter a Character: $
It is some other Character!
Do you want to enter again(Y/N): N
```

Task 9: Encryption/Decryption

Write a program which ask the user to enter a message. Then you have to store that message in a string variable(**msg**). Then program ask the user to enter a positive integer(**k**). Then you have to encrypt the message by adding the k in each alphabet of the **msg** and store the encrypted message in string variable(**encryptedMsg**). After that you have to decrypt the message stored in **encryptedMsg** by subtracting the k in each alphabet of the **encryptedMsg** and store the decrypted message in string variable **decryptedMsg**. At the end you have to print the contents of **encryptedMsg** and **decryptedMsg**.

Hint: Use a loop.

1) You can access single character of string by using subscript operator [ ]
Memory representation of string str = **"Hello World";** is as follows.

0	1	2	3	4	5	6	7	8	9	10
H	e	l	l	o		W	o	r	l	d

If you want to access first character of Hello World then the syntax is cout << str[0]; //H

If you want to access fifth character of Hello World then the syntax is cout << str[4]; //o

2) If you want to add something in a character then the syntax is
   cout <<str[0]+2; //J
   Cout << str[4]+2; //n
3) You can store something in string variable by using the compound assignment operator +=
   string str1=”Ab”;
   cout<< str1; //Ab
   str1 +=str[1]; cout<< str; //Abe
4) You can get the total number of characters in a string by using the length() function.
   int I = str.length(); cout << i; //11

Sample Run:

```
Enter a Message: Hello World!!
Enter a positive Integer: 2

Encrypted Message: Jgnnq"Yqtnf##
Decrypted Message: Hello World!!
```

Task # 10: a^nb^n

Write a program which ask the user to enter a string of combinations of a's and b's. Then you have to print whether the entered string is valid $\mathbf{a^nb^n}$ string or not.

Hint: A string is said to be a valid $\mathbf{a^nb^n}$ if number of a's is equal to the number of b's in string and also all a's comes before all b's.
Assumption: You can assume user can always enter a string consisting of only a's and b's.

Sample Run:

```
Enter a String:  aaaabbbb
It's a valid string.
```

```
Enter a String: aabbba
It's an invalid string.
```

```
Enter a String:  abb
Its's an invalid string.
```

```
Enter a string: bbbaaa
It's an invalid string.
```

Task 11: CGPA Calculator

Write a program which ask the user to enter his full name then his roll number (integer value) then his college name and at last ask his number of semesters completed. Now ask the user to enter GPA of the given number of semesters. Calculate the CGPA and displays it as shown in the sample output.

Sample Run:

```
Enter your Name: Dr. Watson
Enter your Roll Number: 12
Enter Your College Name: Punjab University College of
Information Technology
Enter Number of Semesters: 2
Enter Your Semester 1 GPA: 3.5
Enter Your Semester 2 GPA: 3.8
********************* Result Card***************
Dr. Watson
Punjab University College of Information Technology
CGPA: 3.65
```

Note: Assume that Semester 1 has Credit hours 12, Semester 2 has 13, Semester 3 has 14 and so on. Number of Semesters can't be less than 1 and greater than 8, Roll Number Can't be negative, GPA can't be less than 0 or greater than 4. If any of these constraints are broken indicated error and prompt the user again and again until he enters the value in the correct range.

Task 12: Fist bump Counter

There's a Group of friends in which whenever they meet as a whole group, they all do fist bump with each other. If everyone in the group does fist bump exactly one time with every other group member. How many fists bump is there?
For example: If there's group of 3(Alex, Justin, Andrea).
Alex fist bump Justin, Justin fist bump Andrea and Andrea Fist bump Alex. So, they all fist bumped each other in total of 3 fist bumps.

Sample Runs:

```
Enter Number of Group Members: 4
Total Fist bumps are 6
```

```
Enter Number of Group Members: 2
Total Fist bumps are 1
```

```
Enter Number of Group Members: 0
Total Fist bumps are 0
```

Task 13: Corresponding Number Finder

In this task you have to write a program which finds out the corresponding number of 1 series to other. Let's say there's a series.

Example:
Series 1:

5,7,9,11,13,15

Here 5 is a1, and the difference between values is d which is 2.
Both a1 and d will be entered by the user.

Series 2:
The first number of Series 2 which is n1 will be entered by the user, if n1=7 then series 2 will be,

7,8,9,10,11,12

Note that Series 2 starts with a user specified value and goes on with the increment of 1.
Now ask the user to enter the number from series 1 and then print the corresponding value from series 2.
If user enters 11 You should print 10 on the screen, for 5 it's 7 and for 15 you should print 12.

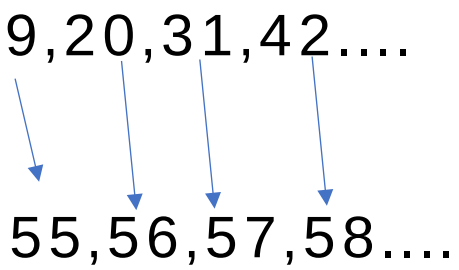
Sample Run 1:

```
Enter First Number of Series 1: 5
Enter the difference of series 1: 2
Enter First Number of Series 2: 7
Enter Number from the Series 1: 13
Your Corresponding Number is 11.
```

Sample Run 2:

```
Enter First Number of Series 1: 9
Enter the difference of series 1: 11
Enter First Number of Series 2: 55
Enter Number from the Series 1: 42
Your Corresponding Number is 58.
```

Explaination:



You can Assume that user will always enter a legal value from series 1 e.g in the sample run above user will not enter 40 as it's not in the series.

Hint:
Make Such series and work on their correspondence.

Task 14: Toggle String

Write a program which ask the user to enter a sentence and store that sentence in a string called (**str**). After that you have to replace all the capital alphabets with small alphabets and small alphabets with capital alphabets. And store that toggled string in a string variable named (**toggledStr**) and print that string on screen.
Note: If the character in the string is not alphabet then you do not have to perform any operation on it.
Hint: Try to find the commonality in ASCII of small and capital alphabets.

Sample Run:

```
Enter a Sentence: Hello World! 123 Check.

Toggled Sentence: hELLO wORLD! 123 cHECK.
```

Task 15: 2D Palindrome

Write a program which ask the user to enter 5 words of length 5. Then you have to display a message whether the square of words is 2D palindrome or not.

Note:
A Words Square is said to be a 2D palindrome if each of the 5 words can be read from bottom to top, top to bottom, left to right, right to left.

Hint: If the string in first row is equal to the reverse of fifth row string and string in the second row is equal to the reverse of fourth row string and the string in third row is equal to the reverse of third row string. And also, if the string in first column is equal to the reverse of fifth column string and string in the second column is equal to the reverse of fourth column string and the string in third column is equal to the reverse of third column string then we say Square is 2D palindrome.
Assumption: The user always enters a world of length 5.

Sample Run:

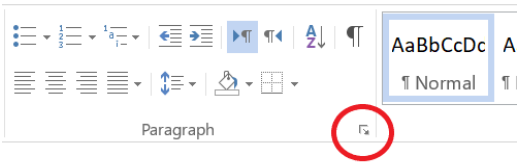
```
Enter 1st String: SATOR
Enter 2st String: AREPO
Enter 3st String: TENET
Enter 4st String: OPERA
Enter 5st String: ROTAS

The Square is a 2D Palindrome!!
```

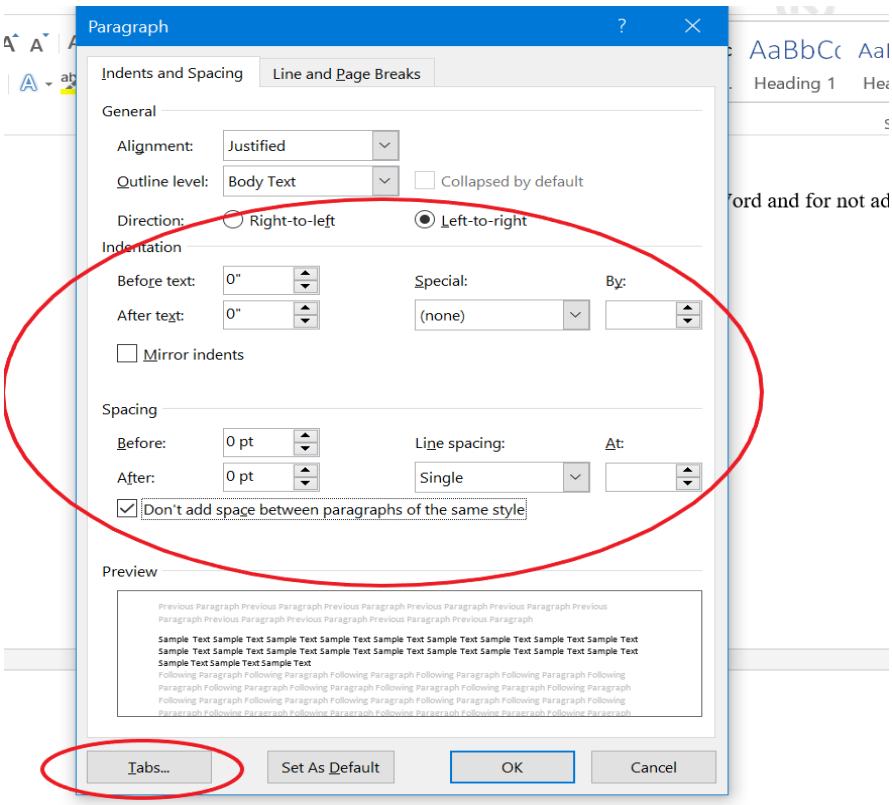
Submission Procedure

You have to paste all your tasks in one MS Word document. Give heading Task 1 and Paste it's solution under it and so on.

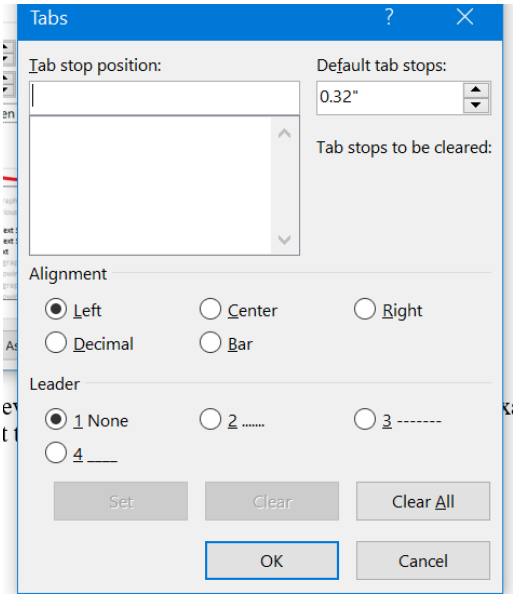
- Use font Menlo or Verdana, Font size 9.
- In order to add tab for the purpose of indentation (a standard tab is of 4 spaces), your tab size should be of 4 spaces.
- Don't use extra spacing between lines of code.
- Following images may help you set the default tab size in MS Word and for not adding extra spacing between lines.



A new window will open after this.



Apply same settings to prevent any extra line spacing. Everything should be exactly same.
Ann then open Tabs to set tabs settings.



- Set Default Tab Stops to 0.32” or 0.8 cm. All other settings should be same as above image.
- Press Ok.
- You have to do this line spacing settings only first time you open the document.
- Then Paste all your Solutions with proper headings,
- Convert the Word File in pdf format and named as **BITF20M001-A2.pdf** (change according to your roll no) and submit through google classroom**.**
- You can easily convert this word file in pdf format by doint “Save as” pdf.

Good Luck