


National University of Computer and Emerging Sciences, Lahore Campus

	Course Name:	Programming Fundamentals	Course Code:	CS
	Program:	CS	Semester:	Fall 2019
	Duration:	60 Minutes	Total Marks:	20
	Paper Date:		Weight	
	Section:	ALL	Page(s):	2
	Exam Type:	Sessional - I		

Student : Name: _____ Roll No. _____ Section: _____

Instruction/Notes: Solve the exam on this question paper. You may use rough sheets, but they must not be attached.

Problem 1 [5 pts] The following C++ code is designed to take an integer Number as input and it should compute and print total number of years, months, weeks and days, possible in this Number. Assume there are 365 days in a year and 30 days in a month.

For example, if the input number is **800**, then it should print,

In Number 800 there are 2 Years 2 Months 1 Weeks and 3 Days.

If the input number is **370**, then it should print,

In Number 370 there are 1 Years 0 Months 0 Weeks and 5 Days.

Unfortunately, the program is not working correctly. Identify the **logical errors** in the code below **by circling them**, and provide corrections in the corresponding line based on the above description.

code	Corrections (only)
y, m, w, d, a, number	
PRINT "Enter a Number: "	
INPUT number	
y = number / 365	
a = number / 365	
m = a / 30	
a = a % 30	
w = a % 7	
d = a % 7	
PRINT "In Number " , number , " there are "	
PRINT y , " Years " , m , " Months " , w , " Weeks and " , d , " Days."	

Problem 2 [5 pts] Give the output of the following code for different inputs of variable extra in the box provided.

```
extra = 0
READ extra;
if (extra < 0){
    PRINT "small"
}
else if (extra is 0){
    PRINT "medium";
}
else{
    PRINT "large";
}
```

extra	Output
2	
-38	
0	

Problem 3 [10 pts] Write a pseudo-code that prints the first n terms of the Tick-Tock series. The value of n is taken as input. The first 8 terms of the Tick-Tock series, i.e., for $n=8$ are as follows:

$$2 - 5 + 4 - 10 + 6 - 15 + 8 - 20$$

Can you guess the pattern? The odd terms are multiples of 2 and in increasing order (2, 4, 6, 8 etc.). The even terms are increasing multiples of 5 and in negative form. The terms are alternatively positive and negative. Nothing should be printed if the value of n is below 1.

For Example:

If the input is $n=5$, the pseudo-code prints: $2 - 5 + 4 - 10 + 6$

If input is $n=10$, the pseudo-code prints: $2 - 5 + 4 - 10 + 6 - 15 + 8 - 20 + 10 - 25$