

Department of Computer Science and Engineering  
Assignment 3  
**Subject : Programming and Data Structure (CS19003)**

---

**Instructions:**

- Give the name of the programs files as <Your roll>.<assignment number>.<question number>.c. For example, 21XX12345\_A1\_Q1.c for question 1 of assignment 1 of a student with roll 21XX12345. **Follow the file naming convention strictly.**
  - Apart from the main .c file for each program, you should also upload one additional temporary .c file for each program (such as when you have finished half of the code). The naming for the temporary file should be in the format <Your roll>.<assignment number>.<question number>.temp.c. For example, 21XX12345\_A1\_Q1.temp.c **Make sure that your main code do not deviate much from its temporary code for each program.**
  - You should upload the main .c file and the temporary .c file individually to the Moodle course web page once you finish answering each question. No need to zip the files.
  - The **deadline** to upload the programs is 12:00PM. Beyond that, submission will be closed (No extensions will be granted).
  - If you do not follow the instructions, your marks may be deducted.
- 

**Answer Part A (compulsory) and any two questions from Part B.**

[20 + 40 + 40 = 100 Marks]

**Part - A**  
**[Compulsory]**  
**[20 marks]**

1. You have 'n' number of balls numbered from 1 through n (1,2,3,.....,n). You are playing a game where you select any two cards each of which has some integer number printed over it. Your task is to choose only those balls whose associated numbers are a multiple of either of the two numbers printed on the cards that you have selected (i.e any one or both the card numbers). Write a program for this. Your input should be three integer numbers, 'n' representing the number of balls and 'x' and 'y', each representing one of the two numbers on the cards that you select. Your output should be those balls (i.e the number associated with the balls) that satisfy the game criteria. Print "Game Lost" in case no balls can be selected. You can assume all positive integer values including 1 but excluding 0 for 'n', 'x' and 'y'.

**Example 1:**

**Input:**

n = 20

x = 4

y = 7

**Output:**

4  
7  
12  
14  
16  
20

[Reason: The numbers on the output balls all satisfy the condition that they are either a multiple of 4 or 7 or both.] [20]

**Part - B**  
**[Attempt any two questions]**  
**[80 marks]**

1. Take input 3 numbers from the keyboard. Find out the number of digits of each of the three numbers. Hence find out whether you can construct a triangle whose three sides have length equal to the number of digits of the three numbers. Also if a triangle is possible, then comment on its type based on the length of its side, i.e whether it is a scalene, isosceles or equilateral triangle. You should display i) the number of digits of each of the three numbers that you have taken as input, ii) print 'Yes' if you can form a triangle with sides equal to the number of digits, 'No' otherwise and iii) display 'Equilateral', 'Isosceles' or 'Scalene' based on the nature of triangle, if you can form the triangle, else print 'Triangle not possible'. To include more digits within a given number, you can assume 'long int' type data while taking input. You may further assume that the input numbers will always fall within the range of 'long int'.

**Example 1:****Input:**

52  
286325  
74523

**Output:**

2  
6  
5  
Yes  
Scalene

[Reason: The first three outputs show the number of digits of the three input numbers. You can form a triangle with sides 2,6 and 5, which is a scalene triangle.]

**Example 2:****Input:**

1  
23  
7564

**Output:**

1  
2  
4  
No

Triangle not possible

[40]

2. Consider the following series of numbers, **4,8,14,22,32 . . . .** This series follows a specific pattern. Find out the pattern. Hence take a number 'k' as input and print the first k terms of the series according to the rules of the pattern. Also take another integer 'n' as input and print the difference between the (n+1)'th term and the n'th term of the series. **You must write the logic of the pattern as well as the general formula for its n'th term as comments below your code. You can assume that n starts from 1.** [40]
3. Consider the infinite summation formula for calculating the value of 'x' as given below. Take two integer numbers 'n' and 'k' as inputs. Find out the value of 'x' by summing the first 'n' terms according to the formula correct to three decimal places. Hence, multiply the value of x with 'k' and consider the floor of the result, i.e the integer value immediately lower than the multiplication value, and check whether the floor value is a prime number or not. You have to display the value of 'x' correct up to three decimal places and print 'Prime' or 'Not prime' based on the primality of floor(x\*k). You can calculate the floor value by any way you feel like. Also make sure to take appropriate variable data types. Read the question clearly, you will get all the necessary details about data types.

$$x = \frac{(1 \times 2)^3}{((1 \times 2)(1 + 2))^2} + \frac{(2 \times 3)^3}{((2 \times 3)(2 + 3))^2} + \frac{(3 \times 4)^3}{((3 \times 4)(3 + 4))^2} + \dots \quad (1)$$

**Example 1:**

**Input:**

n=2  
k=677

**Output:**

0.462  
Not prime

[Reason: As n=2, the value of 'x' as per the given formula considering only the first two terms correct upto three decimal places is 0.462. Now 0.462 x k=0.462 x 677=312.774. Consider the floor of the answer, i.e 312, which is not a prime number.] [40]

4. Write a program to perform the following tasks,
- (A) Take input two positive integer numbers 'n' and 'k'.
  - (B) Hence successively keep on taking positive integer numbers as input until you encounter 'k' multiples of 'n', i.e all the various different multiples of 'n' appearing for a total of 'k' times in the input stream.

(C) Find out the maximum and the minimum among those 'k' multiples of 'n' and display them.

**You have to use a do-while loop for this question and cannot use arrays. Do not consider 0 or negative numbers as inputs anywhere.**

**Initial Input:**

n=5

k=3

**Successive Inputs:**

34

23

25

78

100

32

11

85

**(STOP HERE, NO MORE INPUT)**

**Output:**

Maximum multiple: 100

Minimum multiple: 25

[Reason: As n=5 and k=3, you would need to stop taking successive integers as input once you encounter any 3 multiples of 5. In the given input stream, the 3 multiples of 5 are 25, 100 and 85, in that order. You stop taking further input once you encounter a total of 3 multiples of 5, i.e 85 in this case. Among the three multiples, 25, 100 and 85, 25 is the minimum and 100 is the maximum]

**[40]**