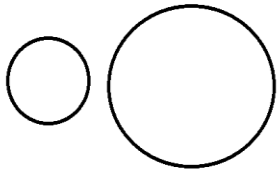
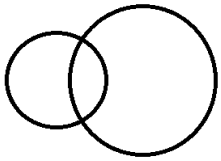
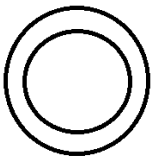
Rules

1. Submission Steps:
   a. In your local machine, create a new folder; the name of the folder should be your 7 digit roll number (e.g. 19050XX).
   b. For each problem that you choose to solve, code your solution in a .c file. The name of the file should be exactly as is mentioned in problem description (e.g. for problem 1 filename will be **circle.c**). Put this C file in the folder created in step 1(a).
   c. Finally, zip the folder created in step 1(a) to produce a .zip file. The name of the .zip file should be your 7 digit roll number (e.g. 19050XX.zip).
   d. Submit the .zip file created in step 1(c).
2. You must submit your solution package (.zip file) by 06:00AM 15/03/2020 through Moodle.
3. You must be able to explain your code properly in the lab class.
4. Do not copy code.

Problem #1
Triangle Classification
File name: circle.c

Take as input 6 integers: $x_1, y_1, r_1, x_2, y_2, r$. (x_1, y_1) represents the center and r_1 represents the radius of circle C_1. The other three integers represent another circle C_2 the same way. Find the status of the two circles as follows:

-C_1 and C_2 **do not intersect**	
-C_1 and C_2 **intersect** (including touch)	
-One is inside the other	

Sample Input	Sample Output
0 0 1 5 0 2	Do not intersect
0 0 4 5 0 3	Intersect
0 0 5 0 0 2	One is inside the other

Problem #2

Sum of Digits

File name: digit_sum.c

Write a C program which takes as input a positive integer number and prints the sum of digits which are prime numbers (Assume that 1 is not a prime number).

Sample Input	Sample Output
1234	5
77	14
347	10

Problem #3

Income Tax

File name: tax.c

Assume income tax of a person is calculated using the rule in the table below. Write a program which takes as input the income of a person and prints the amount of tax imposed on him. **Note: You have to use switch case to solve this problem.**

Income	Tax
0 <= income < 10000	10% of the total income
10000 <= income <30000	10% of 10000tk + 15% of the rest
30000 < income <50000	15% of 30000tk + 20% of the rest
income >= 50000	20% of 50000tk + 25% of the rest

Sample Input(s)	Sample Output(s)
9000.0	900.0
40000.0	6500
12345.67	1351.8505

Problem #4

Series

File name: series.c

In this problem, your input is a real number *x,* which represents an angle in **degree**. You need to apply the following series expansion to compute the value of a series: $\boldsymbol{series}(\boldsymbol{x})$. Use the first 100 items from the infinite series. Print 6 digits after the decimal point in your output. The expansion of the series is as follows (where x is expressed in **radian**):

$$\text{series}(x) = \sum_{n=0}^{\infty} \frac{(-1)^n}{(2n)!} x^{2n} = 1 - \frac{x^2}{2!} + \frac{x^4}{4!} - \frac{x^6}{6!} + \cdots \infty$$

(Hint: Notice that the *n*th term is obtained from the *(n-1)*th term by multiplying with $\frac{-x^2}{(2n-1)*2n}$)

See the following example (minor precision error would be acceptable):

Sample Input(s)	**Corresponding Output(s)**
60	0.500000
85	0.087156
0	1.000000
-210	-0.866025

Problem #5

Longest Positive Sequence

File name: lps.c

In this problem you will be given a sequence of single digit numbers as input. The input will contain both positive and negative numbers and the sequence will stop when 'q' is pressed. You have to find the length, sum and starting position of the longest positive sequence. A positive sequence means a sequence of consecutive positive numbers. **You cannot use array for this problem**.

Sample Input: 1 2 4 -3 -2 1 9 -3 -2 -9 4 5 2 9 -1 -4 q

Sample Output: Length of LPS: 4

Sum of LPS: 20

Starting position of LPS: 10 (Assuming that position starts from 0)