Problem 1.
Short answers.
(1) Assume:

```
double x = 2.5;
double y = -1.5;
int m = 10;
int n = 4;
```

What is the value of the following four expressions?

```
3 * x - n / 5
!(n > m)
(m > n) || (x==y)
(m > n) && (x==y)
```

Line 1: 6.7
Line 2: True
Line 3: True
Line 4: False

(2) what is the output of the following block of C ++ code?

```
int i = 5, j = 6, k = 7, n = 2;
float b = 2.0;
cout << ++j << endl;
cout << i + j * k - k % n << endl;
cout << i / n << endl;
cout << i / b << endl;
```

7
53
2
2.5

(3) What is the value of the variable letter after running following two statements?

```
char letter = 'c';
Letter++;
```

d

Problem 2. Write a C++ program that asks the user to enter five numbers: r, x0, y0, x1 and y1. The first three numbers define a circle centered at (x0, y0) with radius r in 2-D space. The last two numbers represent a point (x1, y1) in 2D space. The program should output “The point is

within the circle." if the distance between the point (x1, y1) and the center of the circle (x0, y0) is less or equal to r, otherwise, the program should output "The point is outside the circle".

Please test the case x0 = 0, y0 = 0, r =1, x1 = 1.0 and y1 = 0.5 and report your output in your write-up.

My Output:

What is the x-coordinate for the center of the circle?
0
What is the y-coordinate for the center of the circle?
0
What is the radius of the circle?
1
What is the x-coordinate of the point you want to determine is within the circle?
1.0
What is the y-coordinate of the point you want to determine is within the circle?
0.5
The distance between the 2 points is 1.11803.
The point is outside the circle.

Problem 3. Implement Banker's rounding algorithm: (1) ask the user to enter a number (2) if the fraction is less than 0.5, round down; if the fraction is greater than 0.5, round up; if the fraction is equal to 0.5, round to the nearest even number (3) display the rounded number on the screen.

Please testing the following four numbers 0.4, 0.6, 0.5, 1.5 as input and report what are the output in your write up.

My Output:

student@student-VirtualBox:~/Desktop/Lab-3$ ./RoundResultType in a positive number you would like to use Banker's Rounding on.
0.4
The rounded number is 0
student@student-VirtualBox:~/Desktop/Lab-3$ ./RoundResult
Type in a positive number you would like to use Banker's Rounding on.
0.6
The rounded number is 1
student@student-VirtualBox:~/Desktop/Lab-3$ ./RoundResult
Type in a positive number you would like to use Banker's Rounding on.
0.5

The rounded number is 0
student@student-VirtualBox:~/Desktop/Lab-3$ ./RoundResult
Type in a positive number you would like to use Banker's Rounding on.
1.5
The rounded number is 2

Problem 4. The Gregorian reform modified the Julian calendar's scheme of leap years as follows: every year that is exactly divisible by 4 is a leap year, except for years that are divisible by 100 but not divisible by 400. For example, the years 1700, 1800, and 1900 were not leap years, but the years 1600 and 2000 were leap years. Complete the following C++ program to assign true to the Boolean variable isLeapYear if the integer variable year is a leap year.

Test your results using year 1700, 1701, 2020 and 2000 and report your output in the write-up.

```
#include <iostream>
using namespace std;
int main (){

int year;
bool isLeapYear;
cin >> year; // you can assume this is a valid input

[Your code here]

if (isLeapYear){
  cout << year << " is a leap year." << endl;
}else{
  cout << year << " is not a leap year." << endl; }

return 0;
}
```

My Output:
student@student-VirtualBox:~/Desktop/Lab-3$ g++ -o YearTest LeapYear.cpp
student@student-VirtualBox:~/Desktop/Lab-3$ ./YearTest
What year would you like to see is a leap year?
1700
1700 is not a leap year.
student@student-VirtualBox:~/Desktop/Lab-3$ ./YearTest

```
What year would you like to see is a leap year?
1701
1701 is not a leap year.
student@student-VirtualBox:~/Desktop/Lab-3$ ./YearTest
What year would you like to see is a leap year?
2020
2020 is a leap year.
student@student-VirtualBox:~/Desktop/Lab-3$ ./YearTest
What year would you like to see is a leap year?
2000
2000 is a leap year.
```