Lab Questions

1. Create a program that takes an integer as input and prints `Positive`, `Negative`, or `Zero` based on its sign.

2. Write a program that asks the user to enter a month and year and then displays the number of days in that month. Account for leap years for February.

3. Write a program `roll_dice.cpp` that generates the result of rolling a fair six-sided dice (an integer between 1 and 6).

4. Write a program `hurricane.cpp` that that takes the wind speed (in miles per hour) as integer input and prints whether it qualifies as a hurricane, and if so, whether it is a Category 1, 2, 3, 4, or 5 hurricane. Below is a table of the wind speeds according to the *Saffir-Simpson* scale.

Category	Wind Speed (mph)
1	74 - 95
2	96 - 110
3	111 - 130
4	131 - 155
5	156 and above

5. *Quadratic equations:* The two roots of a quadratic equation can be obtained using the following formula:

$$r_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a} \quad \text{and} \quad r_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

where $b^2 - 4ac$ is called the discriminant of the quadratic equation. If it is positive, the equation has two real roots. If it is zero, the equation has one root. If it is negative, the equation has no real roots.

Write a program `quadratic.cpp` that prompts the user to enter values for `a`, `b`, and `c` and displays the result based on the discriminant. If the discriminant is positive, display two roots. If the discriminant is 0, display one root. Otherwise, display "`The equation has no real roots.`"

6. Write `lottery.cpp` to play lottery. The program randomly generates a lottery of a three-digit number, prompts the user to enter a three-digit number, and determines whether the user wins according to the following rules:

   1. If the user input matches the lottery number in the exact order, the award is 1 million rupees.
   2. If all digits in the user input match all digits in the lottery number, the award is 5 lac rupees.
   3. If one digit in the user input matches a digit in the lottery number, the award is 1 lac rupees.

7. *Day of the week:* Write a program `day_of_week.cpp` that takes a date as input and prints the day of the week that date falls on. Your program should take three command-line arguments: `m` (month), `d` (day), and `y` (year). For `m` use 1 for January, 2 for February, and so forth. For output print 0 for Sunday, 1 for Monday, 2 for Tuesday, and so forth. Use the following formulas, for the Gregorian calendar (where / denotes integer division):

$$\begin{aligned} y_0 &= y - (14 - m)/12 \\ x &= y_0 + y_0/4 - y_0/100 + y_0/400 \\ m_0 &= m + 12 * ((14 - m)/12) - 2 \\ d_0 &= (d + x + 31m_0/12) \mod 7 \end{aligned}$$

For example, on which day of the week was August 2, 1953?

$$\begin{aligned} y_0 &= 1953 - 0 = 1953 \\ x &= 1953 + 1953/4 - 1953/100 + 1953/400 = 2426 \\ m_0 &= 8 + 12 * 0 - 2 = 6 \\ d_0 &= (2 + 2426 + (31 * 6)/12) \mod 7 = 2443 \mod 7 = 0 \end{aligned}$$

The answer is 0, which means Sunday.

8. *Sum of Numbers:* Create a program that asks the user for a positive integer `n` and calculates the sum of all integers from 1 to `n`. Use a *while* loop for the summation and display the result.

   Hint: Use a `while` loop as follows:

```
int sum = 0;
int i =  __ ;
while ( __________ ) {
    sum = sum + i; // add i to sum
    i =  __________ ;
}
```

9. *Multiplication Table:* Create a program that generates the multiplication table for a given integer `n` up to 10. Use a *while* loop for the multiplication and display the result.

   Hint: Use a `while` loop as follows:

```
int i = 1;
while ( __________ ) {
    cout << n << " * " << i << " = " << n*i << endl;
    i =  __________ ;
}
```