Programming Worksheets

Write C++ programs to do the following

1. Accept a,b,c from user then gives the roots of If the equation has one solution give it if it has no solutions state that.
2. Write a program that asks the user for a number n and prints the sum of the numbers 1 to n.
3. Calculate e using the approximation + ….1/n! the user will provide n
4. You have an array give the summation of its elements.
5. You have an array give the maximum element in it.
6. You have 2-dimensional array, check if it is magical.
7. Write a program which simulates one or more dice with values from 1 to 6. The program takes a single argument which is the number of dice. The output should contain the values of the dice and the probability for this combination to occur. The probability is expressed as a decimal value between 0 and 1 with three decimal points.
8. Write a program which reads a number of strings from standard input and prints them in reverse order on the screen. The input sequence is terminated with the string END.
9. Write a program which reads strings from standard input and sorts them using the Bubble sort algorithm. Determine the execution time with 10, 1000, 1000000 entries using time.
10. Write a program that checks if the number prime or not.
11. Write a program that prints all primes up to the largest number you can represent.
12. Write a guessing game where the user has to guess a secret number. After every guess the program tells the user whether his number was too large or too small. At the end the number of tries needed should be printed. I count only as one try if the user inputs the same number consecutively.
13. Write function that reverses a list, preferably in place.
14. Write a function that checks whether an element occurs in a list.
15. Write three functions that compute the sum of the numbers in a list: using a for-loop, a while-loop and recursion.
16. Write a function that merges two sorted lists into a new list.
17. Write a function that takes a number and returns a vector of its digits.

Write a program that calculates Conway’s game of life . The universe of the Game of Life is an infinite two-dimensional [orthogonal](https://en.wikipedia.org/wiki/Orthogonal) grid of square *cells*, each of which is in one of two possible states, *alive* or *dead*. Every cell interacts with its eight [*neighbours*](https://en.wikipedia.org/wiki/Moore_neighborhood), which are the cells that are horizontally, vertically, or diagonally adjacent. At each step in time, the following transitions occur:

1. Any live cell with fewer than two live neighbours dies, as if caused by under-population.
2. Any live cell with two or three live neighbours lives on to the next generation.
3. Any live cell with more than three live neighbours dies, as if by overcrowding.
4. Any dead cell with exactly three live neighbours becomes a live cell, as if by reproduction.

The initial pattern constitutes the *seed* of the system. The first generation is created by applying the above rules simultaneously to every cell in the seed—births and deaths occur simultaneously, and the discrete moment at which this happens is sometimes called a *tick* (in other words, each generation is a pure function of the preceding one). The rules continue to be applied repeatedly to create further generations.

|  |  |
| --- | --- |
|  |  |
|  |  |