Assignment - 3

Question – 1

a) In this problem, you will utilize the IEEE 754 format and evaluate the performance implications of using floats versus doubles in a computation.
Compute f(x) = sin (x) using a Taylor series expansion. To refresh your memory:

$$\sin(x) = \sum_0^\infty \frac{(-1)^n}{(2n+1)!}\, x^{2n+1}$$

$$\sin(x) = x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} \ldots .$$

You should select the number of terms you want to compute (but at least 10 terms). Compute sin(x) for different values, though be careful not to use too large a value. Generate two versions of your code, first defining x and sin(x) to use floats (SP), and second, defining them as doubles (DP). Discuss any differences you find in your results for f(x). You should provide an in-depth discussion on the results you get and the reasons for any differences.

b) Provide both IEEE 754 single and double precision representations for the following numbers: 2.1, 6300, and -1.044

Answer:

```
Comparison of sin(x) with 10-term Taylor expansion:
---------------------------------------------------
   x       float_Taylor        double_Taylor       std::sin(x)
---------------------------------------------------
0.100000001       0.099833414       0.099833417       0.099833417
1.000000000       0.841470957       0.841470985       0.841470985
3.140000105       0.001592324       0.001592652       0.001592653
6.280000210      -0.004228009      -0.004222460      -0.003185302

IEEE 754 Representations:
-------------------------
Value = 2.100000000
  Single Precision (32 bits): 01000000000001100110011001100110
  Double Precision (64 bits): 0100000000000000110011001100110011001100110011001100110011001101

Value = 6300.000000000
  Single Precision (32 bits): 01000101110001001110000000000000
  Double Precision (64 bits): 0100000010111000100111000000000000000000000000000000000000000000

Value = -1.044000000
  Single Precision (32 bits): 10111111100001011010000111001011
  Double Precision (64 bits): 1011111111110000101101000011100101011000000100000110001001001110
```

1 (a):

The results of the Taylor series for sin(x) in single precision (float) and double precision (double) can differ due to the nature of floating-point arithmetic and the characteristics of each precision. These discrepancies are typically related to precision limits, rounding errors, and accuracy, which are readily evident during the computation process. Here are some of the characteristics we look into:

- **Rounding errors**: Each operation in a floating-point computation has the potential to introduce a rounding error. Rounding errors can accumulate in operations such as addition, subtraction, multiplication, and particularly division (as employed in the calculation of each term of the Taylor series) due to the finite number of bits used to represent floating-point numbers. The accumulation of these errors is more apparent in single precision due to the limited number of bits accessible for number representation.

  The Taylor series for sin(x) involves alternating terms of increasing power but decreasing factorial growth in the denominator. As x grows larger, even though the factorial in the denominator grows super-exponentially, intermediate calculations may reach a size where their precision is compromised in a float. In double precision, the effects of these large numbers are mitigated more effectively.

- **Precision and Accuracy**: Single precision offers a precision of approximately seven decimal digits. This limitation is particularly relevant in situations where the Taylor series terms are either extremely small or extremely large, particularly for values of x that are not near zero. As you calculate higher-order terms (large powers and factorials in the denominator), the precision limitations can result in truncation errors, which are the inability to represent minor differences between successive terms in single precision. Double precision provides approximately 15-16 decimal digits of accuracy, facilitating more precise calculations of the series, particularly as the term count escalates. This enhanced precision is essential for preserving accuracy as the manipulated values become more smaller or larger, which is characteristic of a Taylor series expansion for trigonometric functions.

- **Specific examples and computation limitations:**
  Comparisons at x = 0.1, 1.0, 3.14 (roughly π), and 6.28 (nearly 2 π):

  For smaller values such as 0.1 and 1.0, the distinction between single and double precision may not be readily observable in the initial terms, as both precisions may do these calculations with considerable accuracy.

  As x approaches numbers such as 3.14 3.14 and 6.28 6.28, the discrepancies in the higher powers and terms of the series become increasingly evident in single precision. At 6.28, particularly, the series terms encompass elevated powers and larger factorials, rendering them more vulnerable to rounding and truncation inaccuracies in single precision.

- **Usage:** The double precision is used when high accuracy is required and computations involve broad range of values and the results of the computations have significant consequences, and errors could lead to substantial issues. The single precision is used when the memory usage is a concern, especially in environments with limited resources and performance is a critical factor, such as in real-time systems or where processing speed is vital, and the environment supports faster processing with single precision.