1. (20pts) What is the largest positive integer that can be represented in binary with 16 bits?

Ans: The largest positive integer representable with 16 bits is 65535.

2. (20pts) Are the non-terminating number \((0.555\cdots)_{16}\) and \((0.111\cdots)_{4}\) same? Support your answer by converting them to decimal fractions as well as to their binary representation.

Ans:

(a) First, we convert them to decimal fractions as follows:

\[
(0.555\cdots)_{16} = 5 \times 16^{-1} + 5 \times 16^{-2} + \cdots \\
= \frac{5}{16} + \frac{5}{16^2} + \cdots \\
= 5 \times \left\{ \frac{1}{16} \left( 1 - \frac{1}{16} \right) \right\} \\
= (0.3)_{10}
\]

and

\[
(0.111\cdots)_{4} = 1 \times 4^{-1} + 1 \times 4^{-2} + \cdots \\
= \frac{1}{4} + \frac{1}{4^2} + \cdots \\
= \frac{\frac{1}{4}}{1 - \frac{1}{4}} \\
= (0.3)_{10}
\]

Hence, they are the same in term of decimal fraction.

(b) Next, we convert them to binary representation as follows:

\[
(0.555\cdots)_{16} = (0.3)_{10} = (0.0111\cdots)_{2} \\
(0.111\cdots)_{16} = (0.3)_{10} = (0.0111\cdots)_{2}
\]

Hence, they are the same in term of binary representation.

3. (20pts) Under what condition can one convert from base \(b_1\) to base \(b_2\) without error?

Ans: If there exist positive integers \(k \leq b_1\) and \(c\) such that \(b_2^k = cb_1\), then the conversion from a number in base \(b_1\) to base \(b_2\) can be done without error.
4. (20pts) Suggest a stable method for computing
\[ c = \frac{\sin(x)}{x} \]
for small \( x \).

Ans:
\[ \frac{\sin(x)}{x} = 1 - \frac{x^2}{3!} + \frac{x^4}{5!} - \frac{x^6}{7!} + \cdots + \frac{(-x)^{n-1}}{(2n+1)!} + \cdots \]

5. (20pts) (MATLAB)
Consider the following summation
\[ S = \sum_{i=1}^{n} (-1)^i \frac{1}{i^2}. \]

(a) To keep the rounding error small in computing, is it better to add the largest terms first? Or is there some advantage in doing this in inverse order?

(b) One can also compute the summation by
\[ S_p = \text{sum of all the positive terms}; \]
\[ S_n = \text{sum of all the negative terms}, \]
and
\[ S = S_p + S_n. \]

Do you think this alternative can have a smaller or larger accumulated rounding error than the approaches in (a)?

Please use MATLAB to support your answers.

Ans:

(a) In general it is better to add the small numbers first, because for a small sum a comparable relative error gives a smaller absolute error.

(b) This is a poor way of doing the sum since it could result in cancellation of large, like quantities.

6. (20pts) (MATLAB)
Conduct an experiment to determine if the recurrence
\[ f(n+1) = f(n) + \frac{1}{n}f(n-1), \]
\[ f(0) = 0, \]
\[ f(1) = 1, \]
is stable. Compute \( f(50) \) and estimate its accuracy.

Ans:
function output = hw2_6(n)
if n == 0,
    output = 0;
elseif n==1
    output = 1;
else
    output = hw2_6(n-1)+(1/(n-1))*hw2_6(n-2);
end