

Homework 1: CS321-003, Spring 2006

Answer Sheet

1. Check the binary-octal table, we have

$$(45653.127664)_8 = (100\ 101\ 110\ 101\ 011.001\ 010\ 111\ 110\ 110\ 100)_2$$

For conversion to decimal numbers, we need to treat the integer and fractional parts separately. For the integer part:

$$\begin{aligned}(45653)_8 &= 3 \times 8^0 + 5 \times 8^1 + 6 \times 8^2 + 5 \times 8^3 + 4 \times 8^4 \\ &= 3 + 8(5 + 8(6 + 8(5 + 4(8)))) = (19371)_{10}\end{aligned}$$

For the fractional part, we compute

$$\begin{aligned}(.127664)_8 &= 1 \times 8^{-1} + 2 \times 8^{-2} + 7 \times 8^{-3} + 6 \times 8^{-4} + 6 \times 8^{-5} + 4 \times 8^{-6} \\ &= (1 \times 8^5 + 2 \times 8^4 + 7 \times 8^3 + 6 \times 8^2 + 6 \times 8^1 + 4) 8^{-6} \\ &= ((((((1)8 + 2)8 + 7)8 + 6)8 + 6)8 + 7) 8^{-6} \\ &= \frac{44980}{262144} = (.17158508\dots)_{10}\end{aligned}$$

Checking the binary-hexadecimal table, we have

$$(C553E000)_{16} = (1100\ 0101\ 0101\ 0011\ 1110\ 0000\ 0000\ 0000)_2$$

Note that this problem is NOT to convert this hexadecimal number to a decimal number. This hexadecimal number is an IEEE 32 bit representation of a binary number.

The first bit is “1”, so this number is negative. The next 8 bits, $(10001010)_2$, represent the exponent, which is (using the binary-octal table)

$$(010\ 001\ 010)_2 = (212)_8 = 2 \times 8^0 + 1 \times 8^1 + 2 \times 8^2 = (138)_{10}$$

Because of the 127 shift convention, the actual exponent is $138 - 127 = 11$. It follows that the decimal number is

$$-(1.101\ 001\ 111\ 100)_2 \times 2^{11} = -(110\ 100\ 111\ 110)_2 = -(6476)_8 = -(3390)_{10}$$

2. We can see that

$$\text{fl}(xy) = xy(1 + \delta_1), \quad \text{with} \quad |\delta_1| \leq 2^{-24}$$

Because xy is computed first, it follows that

$$\begin{aligned} \text{fl}((xy)z) &= \text{fl}(\text{fl}(xy)z) = \text{fl}(xy(1 + \delta_1)z) = (xy(1 + \delta_1)z)(1 + \delta_2) \\ &= xyz(1 + \delta_1)(1 + \delta_2) = xyz(1 + \delta_1 + \delta_2 + \delta_1\delta_2) \\ &\approx xyz(1 + \delta) \end{aligned}$$

Here $|\delta_1| \leq 2^{-24}$, $|\delta_2| \leq 2^{-24}$, and $|\delta| = |\delta_1 + \delta_2| \leq |\delta_1| + |\delta_2| \leq 2^{-23}$. We ignored them higher order term $\delta_1\delta_2$.

3. The most important point here is to realize that if a number is divided by 2, it is equivalent to move the binary point to the left by one position. Thus, the best way to compute the machine ϵ is to set up a loop to check if the identity $1 + \epsilon = 1$ holds, with initially setting $\epsilon = 1$. If the identity holds, you exits the loop, otherwise you divide ϵ by 2, i.e., by setting $\epsilon = \epsilon/2$. When the loop exits, you get your machine ϵ . Note that, in order to prevent the code from running indefinitely due to coding error, you may want to set a maximum number of iterations for the loop, and check the value of ϵ to see if it is small upon exit.
4. Since the number has a finite representation in the binary number system, its fractional part is of finite length (with nonzeros). We can write its integer and fractional parts as

$$\begin{aligned} x &= \pm \left(\sum_{i=0}^{k_1} a_i 2^i + \sum_{j=1}^{k_2} b_j 2^{-j} \right) \\ &= \pm \left(\sum_{i=0}^{k_1} a_i 2^{i+k_2} + \sum_{j=1}^{k_2} b_j 2^{k_2-j} \right) 2^{-k_2} \\ &= \pm m/2^n \end{aligned}$$

Here $m = (\sum_{i=0}^{k_1} a_i 2^{i+k_2} + \sum_{j=1}^{k_2} b_j 2^{k_2-j})$ and $n = k_2$ are positive integers.

5. You need to write a program to evaluate e^x based on the Taylor expansion. Then

$$f_n(x) = e^x = 1 + x + \frac{x^2}{2!} + \frac{x^3}{3!} + \cdots + \frac{x^n}{n!} + \cdots$$

Your code should have two control mechanisms, one is to stop adding terms when the added term, i.e., when $|\frac{x^n}{n!}|$ is smaller than 10^{-6} , the other is when n is larger than 25. You then compare the computed value of $f_n(x)$ with the computer intrinsic function $\exp(x)$ to see the difference and to compute the relative error.

x	$f_n(x)$	n
0.0	1.0	1
1.0	2.7182818285	10
-1.0	0.3678794412	10
0.5	1.64872 12707	8
-0.123	0.8842636626	5
-25.5	$8.4234637545 \times 10^{-12}$	25
-1776	0	25
3.14159	23.1406312270	17