Arnav Surve
CNIT 176
Lab 08
10/26/2022

Question 1:

a) 439878.97234 * 10^12
   0.44387897234 * 10^18
   07344387
b) ±00001 x 10^-99
c) ± 99999 x 10^99

Question 2:

a) −0.700375x10^−28
   Invalid – exponent underflow
b) ±000001 x 10^-99
c) ± 999999 x 10^99

Question 3:

a) 2918281828
   0.29183 x 10^10
   05929183
b) Format can store from ±00001 x 10^-99 to ± 99999 x 10^99
c) 15929183
d) 0.0000029182818284
   0.29183 x 10^-5
   05429183

Question 4:

a) Positive (+)
b) 250 -127 = 123
c) Positive (+)

Question 5:

a) Positive (+)
b) 122-127 = -5
c) Negative (-)

Question 6:
5636096 for both 4 & 5.

Question 7:

>>>0.1+0.1	0.2
>>>0.1+0.2	0.30000000000000004

>>>0.1+0.7	0.7999999999999999
>>>0.3+0.6	0.8999999999999999
>>>0.2+0.7	0.8999999999999999

When I tried to sum these numbers, the interpreter returned sums that were close approximations to the expected value, but not the exact value. This happened as a result of the nature of the IEEE floating-point standard. The IEEE standard includes methods for rounding basic operations. Although there are an infinite number of integers, most programs store the results of integer computations in 32 bits. According to the Oracle documentation, “given any fixed number of bits, most calculations with real numbers will produce quantities that cannot be exactly represented using that many bits. Therefore, the result of a floating-point calculation must often be rounded in order to fit back into its finite representation”. This rounding error is the result of a feature of floating-point computation.

Additional Examples:

```
>>> 0.000003 + 0.000007
9.999999999999999e-06
>>> 0.7+0.1
0.7999999999999999
```

References

Oracle Docs. (2000, April 5). *Appendix D*. What Every Computer Scientist Should Know About Floating-Point Arithmetic. Retrieved October 26, 2022, from https://docs.oracle.com/cd/E19957-01/806-3568/ncg_goldberg.html