Computer Systems Organization

Assignment 2: Floating Points

Due: Oct. 5, 11:55 PM

1 Code of Conduct

All assignments are graded, meaning we expect you to adhere to the academic integrity standards of NYU Abu Dhabi. To avoid any confusion regarding this, we will briefly state what is and isn't allowed when working on an assignment.

Any documents and program code that you submit must be fully written by yourself. You can, of course, discuss your work with fellow students, as long as these discussions are restricted to general solution techniques. Put differently, these discussions should not be about concrete code you are writing, nor about specific results you wish to submit. When discussing an assignment with others, this should never lead to you possessing the complete or partial solution of others, regardless of whether the solution is in paper or digital form, and independent of who made the solution, meaning you are also not allowed to possess solutions by someone from a different year or course, by someone from another university, or code from the Internet, etc. This also implies that there is never a valid reason to share your code with fellow students, and that there is no valid reason to publish your code online in any form.

Every student is responsible for the work they submit. If there is any doubt during the grading about whether a student created the assignment themselves (e.g. if the solution matches that of others), we reserve the option to let the student explain why this is the case. In case doubts remain, or we decide to directly escalate the issue, the suspected violations will be reported to the academic administration according to the policies of NYU Abu Dhabi (see https://students.nyuad.nyu.edu/campus-life/community-standards/policies/academic-integrity/).

2 Introduction

Write a program in C that computes the following summations:

Result = 1 – ½ + 1/3 – ¼ + 1/5 – 1/6 +………..+1/9999 – 1/10000

The calculations should be done in the following ways:

a) Addition of operands from left to right.
b) Addition of operands from right to left.
c) Separate additions of positive and negative operands, each from **left to right** (add all the positive numbers together and then add all the negative numbers together, for each of them perform the operation from left to right, and then add the two results together).
d) Separate additions of positive and negative operands, each from **right to left**(same as C).

The summations above should be accumulated first as float numbers, then double float numbers. Therefore, the program will print eight different calculations.

3 Implementation

The program must have separate loops and the results should be printed with the largest possible numbers of decimals digits.

The program should run correctly, so that there will be eight possible results:

Result-1: addition from left to right accumulated in float
Result-2: addition from right to left accumulated in float
Result-3: all positives + all negatives from left to right accumulated in float
Result-4: all positives + all negatives from right to left accumulated in float
Result-5: addition from left to right accumulated in double float
Result-6: addition from right to left accumulated in double float
Result-7: all positives + all negatives from left to right accumulated in double float
Result-8: all positives + all negatives from right to left accumulated in double float

Please analyze these eight results and explain the observed differences. For instance, you may try to indicate which results are more accurate, and suggest the advantages and disadvantages of calculating the additions one way or another. Describe your own perspective on what is happening and why.

4 Grading

- The correct version of the program is worth 4 points.
- An essay (1 page maximum) with explanation of the results is worth 1 points.

If you are curious you may implement other ways to calculated the additions to see more results. For instance, you may use different expressions to negate the terms of the summation. The examples below don't count as extras they are just examples.

- Evaluate the differences (if any) in the calculations when writing expressions in other forms, such as: **a = b * (-1)** or **a = (-1) * b** or **(-1.0) * b** or **b * (1.0)** or **a = -b**. Other variations of statements are possible, worth trying, but it may not make any difference.
- What about calculating with long double with 10 bytes floating point instead of 8 bytes?

Note that your solution must be written in the C programming language. In case your code does not work your submission will not be graded.

5 Submission

Please submit one zip file containing your **C** program and the essay with your interpretation of the results in another **PDF** file on Bright space (https://brightspace.nyu.edu/) . Submissions via email are not accepted.

Answers: The results of the different summations in 30 digits should be approximately:

0.693097183059945296917232371458

You may use more digits to show the results. If your results are too much different, there is a bug in the program.

There will be slightly different results, particularly in the final digits, and the idea of the assignment is that the students should explain the differences in term of roundings, truncations, and approximations that happens when dealing with floating points arithmetic calculations in a computer.

You should understand the representation of real numbers in the form (s)M x 2^E, and that real numbers are infinite, but the capacity of representation in the computer is finite, i.e. finite number of bits/bytes for M and E.

In order to help you understand the results and to write the essay please refer to materials available in the internet, provided in the course such as: CSO_03D_floatnumbers, CSO_03E_float-math, CSO_03C_Data-Representation-Tutorial section 4, and the textbook section 2.4.

Please write a concise essay, no verbose, straight to the point.

Good luck!