Taylor Series for Sin

> Yuwei XIAO
>
> CS214 (2023 Spring)

Description

Write a program that takes an input floating-point number and calculates the sine of that number using the Taylor Series expansion. Your program should display the output value.

- The Taylor series expansion for sine is:

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

We will first give you a number op, and then give you a **floating-point** number. You should first judge whether op is an integer or a floating-point. If op is an integer, you should regard it as the number of terms in your Taylor Series Expansion calculation. And if op is a floating-point, you should regard it as the tolerance threshold.

After that, we will give you a floating-point number x. You can use the above formula to calculate the sine of a number by adding up the terms of the series until:

- the number of terms equals op (if op is an integer)
- the difference between the latest sum and the previous sum, which equals to the **absolute** value of the new term, is less than op. (if op is a floating-point number)

Then, output the latest sum.

NOTES:

- **ALL NUMBERs in the calculation as well as IO should be taken as float (single-precision). And please use the above methods to calculate.**
- In our case, an integer means its fractional part is ignorable.

  For example, 2.00 is an integer, and 2.01 is a floating-point number.
- The number of terms is calculated by counting the terms in the Taylor Series Expansion.

  For example, if $op = 2$ and $x = 4.5$, then your Taylor Series Expansion should be:

  $sin(4.5) = 4.5 - \frac{4.5^3}{3!} = -10.6875$
- If $op = 2.01$ and $x = 4.5$, then your Taylor Series Expansion should be:

  $sin(4.5) = 4.5 - \frac{4.5^3}{3!} + \frac{4.5^5}{5!} - \frac{4.5^7}{7!} + \frac{4.5^9}{9!} - \frac{4.5^{11}}{11!} = -1.0228913$

  Since $\frac{4.5^9}{9!} = 2.085209 > 2.01$ and $\frac{4.5^{11}}{11!} = 0.383868 < 2.01$, therefore $\frac{4.5^{11}}{11!}$ is the first term which value within our threshold, so the iteration stops.

Hint:

You can read op as a **float**, and use instructions related to **floating-point** to judge it.

To make sure the precision of your result is the same as our, you should set the constant like this:

```
.data
tolerance: .float 0.00001
unit: .float 1.0
factor: .float 1.0
sign: .float -1.0
```

Sample inputs and outputs

Input #1

```
0.00001
1
```

Output #1

```
0.84147096
```

Input #2

```
5.0
1
```

Output #2

```
0.84147096
```

Input #3

```
2.01
4.5
```

Output #3

```
-1.0228913
```

Input #4

```
2.00
4.5
```

Output #4

```
-10.6875
```