* generate\_result(x) : handles the case when x is 0 , by limit theorem answer is 1 . In the other case , we return sin(x)/x.
* sin(x) : function calculate sine of x where x is in degree.
* First we convert x from degree to radians so that we can apply Taylor series expansion to compute the value of sin(x)
* For converting degree to radian , first we convert x in range [0,360) then multiply it with pi/180. Since we cannot use any library, I decided to use the value of pi as 3.14159265359 , taken from the internet.
* Taylor series of sin(x) = x - x^3/3! + x^5/5! - x^7/7! + …
* We initialise the ‘result’ and ‘term’ as x
* We will use ‘term’ for calculating the next term in the taylor series.
* As we can observe next term is always ‘term’ \* -1 \* *x* \* x / ( (2\*i) \* (2\*i+1) ) where i represents the term number (position from left , 0-index)
* Simply , term[i]=term[i-1] \* -1 \* x \* x / ( (2\*i) \* (2\*i+1) ) , where i starts from 1 and term[0]=x
* So, until the absolute value of ‘term’ becomes less than 1e-10 , we modify ‘term’ as mentioned above and add it to the result.
* Finally we return result

*def* sin(*x*):

"""

Computes the sine of x in degrees using the Taylor series expansion

sin(x) = x - x^3/3! + x^5/5! - x^7/7! + ...

"""

*x* =( (*x* % 360) + 360 ) % 360 # convert to range [0, 360)

*x* = *x* \* (3.14159265359 / 180) # convert to radians

result = term = *x*

i = 1

while abs(term) >= 1e-10:

# compute next term in series

term \*= ((-1 \* (*x*\*\*2)) / ((2 \* i) \* (2 \* i + 1)))

# add term to result

result += term

# increment counter for next term in series

i += 1

return result

*def* generate\_result(*x*):

"""

Computes the value of F(x) = sin(x)/x

"""

if *x* == 0:

return 1 # limt as x -> 0

else:

return sin(*x*) /*x*