P11. Given Taylor series, a value of sine function can be approximated from

$$sin(x) \approx \sum_{n=0}^{M-1} \frac{(-1)^n}{(2n+1)!} \cdot x^{2n+1},$$

where M is a number of terms. The larger M is, the more accurate the approximate is. For example,

$$M = 5, sin(x) \simeq x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!}.$$

$$M = 6, sin(x) \simeq x - \frac{x^3}{3!} + \frac{x^5}{5!} - \frac{x^7}{7!} + \frac{x^9}{9!} + \frac{x^{11}}{11!}.$$

Write a program to approximate a sine value: ask a user for x for an angle in radian and M for a number of terms, compute a value of sin(x) for M terms, and report the computation.

Use the P11 template. (P11_template.py. The template is only to ensure the exact display format and allows smooth auto-grading.)

Example:

```
==============================================================
x:0.7

M:2

sin(0.70) = 0.64283
==============================================================
x:0.7

M:3

sin(0.70) = 0.64423

==============================================================
x:0.7

M:4

sin(0.70) = 0.64422
==============================================================
x:0.7

M:5

sin(0.70) = 0.64422
==============================================================
```

Here is P11_template.py

```python
"""
Given Taylor series, a value of sine function can be approximated from
sin(x) \approx sum_{n=0}^{M-1} (-1)^n)/((2n+1)!) x^(2n+1),
where M is a number of terms. The larger M is,
the more accurate the approximate is. For example,
M = 5, sin(x) ≃x-x^3/3!+x^5/5!-x^7/7!+x^9/9!.
M =6, sin(x)≃x-x^3/3!+x^5/5!-x^7/7!+x^9/9!+x^11/11!.
Write a function, named approx_sin(x, tol),
to take 2 argument: x for an angle in radian and tol for tolerance.
The function computer approximate value of sin(x) for M terms such that
the last term in the series is smaller than the tolerance.
"""

def fact(n):
    f = 1
    for i in range(1,n+1):
        f *= i

    return f


def approx_sin(x, M):
    sinv = 0

    # Write your code here!!!

    return sinv


if __name__ == '__main__':
    x = float(input('x:'))
    M = int(input('M:'))

    print('sin(%.2f) = %.5f'%(x, approx_sin(x, M)))
```