Algorithm: Fourier analysis of square wave

Include predefined class libraries / header files in the program using preprocessor directive #include <iostream> , <cmath> , <fstream>

using namespace std;

Declare input file named “squarewave.inp” using ifstream
Declare output file named “squarewave.out” using ofstream

Class declaration: declare a class “fourier”

Private member declaration:

Data variables:
declare integer type data variables “j,n,N”
declare float type data variables “i,w,y,f,h,pi”

Public member declaration:
Declare member function (1) getdata (2) calculate of type void

Member function definition:
Define member functions getdata using scope resolution operator

“void fourier::getdata()”

Get input from user:

"enter height of square wave (h), no of sine wave harmonics to be included into calculations (n), time (N) "

The input is written in the input file that has already been declared “squarewave.inp”
“input>>h>>n>>N;”

Member function definition:
Define member functions calculate using scope resolution operator
“void fourier::calculate()”

Perform the calculations using for loop

```
pi=4*atan(1);
for(i=0;i<=N;i+=0.001)                      //for loop
{
y=0;                                        //variable initialization
w=2*i*pi;
for(j=1;j<=n;j+=2)                          //for loop
{
y=y+(sin(j*w)/j);                           //sine wave harmonics formula
}
f=(h/2)+(2*h*y/pi);                         //fourier analysis formula

// Output the results of calculations in already declared file “squarewave.out”
output<<i<<" "<<f<<endl;
}
```

Inside the main function of type integer , create an object of class fourier and named sqw. Use this object to call public member functions and terminate the program using return 0 statement

```
int main()
{
fourier sqw;                            //object creation
sqw.getdata();                          //member function calling
sqw.calculate();
return 0;                               //program termination
}
```

Plot a graph f vs i