Algorithm: Lagrange Interpolation

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

using namespace std;

Declare input file "inp.dat" using ifstream
ifstream input("inp.dat");

Class declaration: declare a class “lagrange”

Private member declaration:

Data variables:
declare integer type data variables “i,k,n”
declare float type data variables “x[100],y[100],s,z,t”

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

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

“void lagrange::getdata()”

Get input from user:

cout<<"enter number of values (input file)"<<endl;
The input is written in allocated memory space using
input>>n;

cout<<"enter the value at which interpolation in to carried out (input file)"<<endl;
The input is written in allocated memory space using
input>>z;

cout<<"enter data points (input file)"<<endl;
The input is written in allocated memory space using for loop
for(i=1;i<=n;i++)
{
input>>x[i]>>y[i];
}
}

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

Perform calculations using for loop

for(k=1;k<=n;k++)

```
{
t=1;
for(i=1;i<=n;i++)
 {
   if(i!=k)                                          //if  condition loop
     {
     t=t*(z-x[i])/(x[k]-x[i]);                       //calculate coefficients
     }
  }
s=s+(t*y[k]);                                        //Lagrange Interpolation formula
}
cout<<"LGI at "<<z<<" = "<<s<<endl;
}
```

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

```
int main()
{
lagrange lgi;                                        //create object
lgi.getdata();                                       //member function calling
lgi.calculate();
return 0;                                            //program termination
}
```