

### Algorithm: Least Square Fitting

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

using namespace std;

Declare input file named “inp.dat” using ifstream

Declare output file named “out.dat” using ofstream

**Class declaration:** declare a class “LSF”

**Private member declaration:**

**Data variables:**

declare integer type data variables “i,n”

declare float type data variables “x[20],y[20],a,b,s1,s2,s3,s4,z[20]”

where x[20],y[20],z[20] are declared as array and can contain upto 21 values.

**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 LSF::getdata()”

**Get input from user:**

"enter no of values"

The input is written in the input file that has already been declared “inp.dat”

“input >>n”

Get input from user for parameters x[i] and y[i]

these parameters are also input from the file “inp.dat” using “for loop”

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

```
{  
    input>>x[i]>>y[i];  
}
```

**Member function definition:**

Define member function calculate using scope resolution operator

“void LSF::calculate()”

Perform the calculations using for loop

```
for(i=1;i<=n;i++)  
{  
    s1=s1+(x[i]*y[i]);  
    s2=s2+x[i];  
    s3=s3+y[i];  
    s4=s4+(x[i]*x[i]);  
}
```

Calculate slope and intercept using formula

$$a = \frac{(n \cdot s_1) - (s_2 \cdot s_3)}{(n \cdot s_4) - (s_2 \cdot s_2)}$$
$$b = \frac{(-a \cdot s_2) + (s_3)}{n}$$

Output the results of calculations in already declared file “out.data” using for loop

```
for(i=1;i<=n;i++)
{
z[i]=(a*x[i])+b;
output<<x[i]<<" "<<y[i]<<" "<<z[i]<<endl;
}
```

Inside the main function of type integer

create an object of class LSF and named msc.

Use this object to call public member functions

and terminate the program using return 0 statement

```
int main()
{
LSF msc;                //creat object
msc.getdata();           //member function calling
msc.calculate();
return 0;                //program termination
}
```

**Plot a graph y[i] vs x[i] and z[i] vs x[i]**