Lagrange Interpolation Method

This program is used to find the f(x) of a function f at particular x using lagrange interpolation . The program is:

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>

float l(float xi,float x[32],int j,int n);    /* function returns l_i(x) */
/**************Main function begins*************/
int main(int argc,char *argv[])
{
  float x[32],xi = 0,f[32],p = 0;    /* x[32] ,f[32 is the array of x and f(x) respectively */
  int n,i = 0,j = 0;    /* i and j are used to maintain count */
  if (argc != 2) {
    printf("Syntax:./lagrange n(n is the no. of values of x)\n");
    exit(1);
  }
  n = atoi(argv[1]);    /* n is the no. of values of x */
  printf("Enter x value:");
  scanf("%f",&xi);    /* xi is the value for which root is to be found */
  /**** The below loop takes x value and f(x) ***/
  while (i != n) {
    printf("Enter x_%d =",i);
    scanf("%f",&x[i]);
    printf("Enter f(x_%d) =",i);
    scanf("%f",&f[i]);
    i++;
  }
  /*** p is the value of the function at xi ***/
  while (j != n) {
    p = p + (l(xi,x,j,n)*f[j]);    /* p is initialized to 0 when j = i p = p_i + li(xi)f(xi) p_i is the
previously obtained p */
    j++;    /* increments count */
  }
  printf("f(%f) = %f\n",xi,p);    /* prints the function value at xi */
}
/**************Main function ends*******************/
float l(float xi,float x[32],int j,int n) {
  float a = 1,b = 1;
  int i = 0;
  while (i != n) {
    if (i == j) {
      i = i + 1;
      if (i == n) {
        return (a/b);
      }
    }
    a = (a*(xi - x[i]));    /* a  will be equal to product of (x - xj) where j != k at l_k */
    b = (b*(x[j] - x[i]));  /* b  will be equal to product of (xk - xj) where j != k at l_k */
    i++;    /* increments count */
```

```
  }
  printf("l[%d] = %f\n",j,a / b);    /* prints l value */
  return (a / b);    /* returns l value */
}
```

when input is:
./lagrange 4 (there are 4 x's and f(x)'s)

X	5	6	9	11
F(X)	12	13	14	16

To find F(10) :
Output:
l[0] = 0.166667
l[1] = -0.333333
l[2] = 0.833333
f(10.000000) = 14.666666

l_i = l[i]
when input is:
./lagrange 4 (there are 4 x's and f(x)'s)

X	1	2	3	4
F(X)	1	8	27	64

To find F(2.5) :
Output:
l[0] = -0.062500
l[1] = 0.562500
l[2] = 0.562500
f(2.500000) = 15.625000