

```

#include <iostream>
#include <fstream>
#include <cmath>

using namespace std;

void readtosentinel (double a[], int& i);
double mean (double a[], int i);
double variance (double a[], int i);
double std_dev (double a[], int i);
int cross (double a[], int i);

ifstream nums ("samplenums.txt");
const int SIZE = 100;

int main()
{
    ifstream nums;
    int nsize;
    double numbers[SIZE];

    readtosentinel (numbers, nsize);

    if (nsize != SIZE)
    {
        double m;
        m = mean (numbers, nsize);
        cout << "Mean: " << m << endl;
        double v;
        v = variance (numbers, nsize);
        cout << "Variance: " << v << endl;
        double d;
        d = std_dev (numbers, nsize);
        cout << "Standard Deviation: " << d << endl;
        double c;
        c = cross (numbers, nsize);
        cout << "Number of zero crossings: " << c << endl;

        return 0;
    }
    else if (nsize == SIZE)
    {
        cout << "***ERROR**" << endl;
        cout << "ARRAY FULL" << endl;
        return -1;
    }
}

void readtosentinel (double a[], int& i)
{
    i = 0;
    nums >> a[0];
    while (a[i] != 9999 && i < SIZE)
    {
        i++;
        nums >> a[i];
    }
}

double mean (double a[], int i)
{
    double total = 0;
    for (int j = 0; j < i; j++)
    {
        total = total + a[j];
    }
    double mean = total/i;
    return mean;
}

double variance (double a[], int i)

```

```
{
    double total = 0;
    for (int j = 0; j < i; j++)
        total = total + a[j];

    double mean = total/i;
    double sums[i];
    for (int j = 0; j < i; j++)
    {
        sums[j] = pow(a[j] - mean, 2);
    }
    double sum = 0;
    for (int j = 0; j < i; j++)
        sum = sum + sums[j];

    return (sum/(i-1));
}

double std_dev (double a[], int i)
{
    double total = 0;
    for (int j = 0; j < i; j++)
        total = total + a[j];

    double mean = total/i;
    double sums[i];
    for (int j = 0; j < i; j++)
    {
        sums[j] = pow(a[j] - mean, 2);
    }
    double sum = 0;
    for (int j = 0; j < i; j++)
        sum = sum + sums[j];

    double var;
    var = sum/(i-1);

    return (sqrt(var));
}

int cross (double a[], int i)
{
    int times = 0;
    for (int j = 0; j < i-1; j++)
    {
        double test = (a[j] * a[j+1]);
        if (test < 0)
            times++;
    }
    return times;
}
```