Program 1. Write a program to calculate mean, median and mode in an individual series.

#include<stdio.h>

float mean(int num, int \*arr){

    float sum = 0;

    for(int i=0; i<num; i++){

        sum += arr[i];}

    float meann = sum / num;

    return meann;}

float median(int num, int \*arr){

    float ans;

    for(int i=0; i<num; i++){

        for(int j=i+1; j<num; j++){

            if(arr[i]> arr[j]){

                int a = arr[i];

                arr[i] = arr[j];

                arr[j] = a;} } }

    if(num % 2 == 0){

        int one = num / 2;

        int two = one - 1;

        float x = arr[one];

        float y = arr[two];

        ans = (x + y) / 2;

    }else ans = arr[((num + 1) / 2) - 1];

    return ans;}

int mode(int num, int \*arr){

    int max\_Value = 0;

    int max\_Count = 0;

    for (int i = 0; i < num; i++){

        int cnt = 0;

        for (int j = 0; j < num; j++){

            if (arr[j] == arr[i])

                cnt++;}

        if (cnt > max\_Count){

            max\_Count = cnt;

            max\_Value = arr[i];}}

    return max\_Value;}

void main(){

    int num;

    printf("Enter the number of observations  ");

    scanf("%d", &num);

    int arr[num];

    printf("Enter the values of the observation\n");

    for(int i=0; i<num; i++){

        scanf("%d", &arr[i]);}

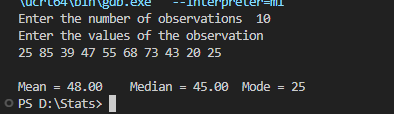
    float Mean = mean(num, arr);

    float Median = median(num, arr);

    int Mode = mode(num, arr);

    printf("\nMean = %.2f\tMedian = %.2f\tMode = %d", Mean, Median, Mode);

}

OUTPUT:

Program 2. Write a program to calculate mean, median and mode in a discrete series.

#include<stdio.h>

void main(){

    int size;

    printf("Enter the number of observations  ");

    scanf("%d", &size);

    int arrX[size];

    printf("Enter the X values\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrX[i]);

    }

    int arrF[size];

    printf("Enter the F values\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrF[i]);

    }

    // Sorting the X and Y table

    for(int i=0; i<size; i++){

        for(int j=i+1; j<size; j++){

            if(arrX[i]> arrX[j]){

                int x = arrX[i];

                int f  = arrF[i];

                arrX[i] = arrX[j];

                arrF[i] = arrF[j];

                arrX[j] = x;

                arrF[j] = f;

    } } }

    // Making of TABLE

    int FX[size], CF[size];

    int add = 0;

    for(int i=0; i<size; i++){

        FX[i] = arrX[i] \* arrF[i];

        add += arrF[i];

        CF[i] = add;

    }

    // Printing the table

    printf("X\tF\tF\*X\tCF\n");

    for(int i=0; i<size; i++){

        printf("%d\t%d\t%d\t%d\n", arrX[i], arrF[i], FX[i], CF[i]);

    }

    //  MEAN

    float sumFX = 0;

    float sumF = 0;

    for(int i=0; i<size; i++){

        sumFX += FX[i];

        sumF += arrF[i];

    }

    float mean = sumFX / sumF;

    printf("\nMean = %.2f\n", mean);

    // MEDIAN

    float term = (sumF + 1) / 2;

    int position = 0;

    for(int i=0; i<size; i++){

        if(CF[i] < term){

            continue;

        }else{

            position = i;

            break;

        }}

    printf("Median = %.d\n", arrX[position]);

    // MODE

    int highest = 0;

    int pos = 0;

    for(int i=0; i< size; i++){

        if(arrF[i] > highest) {

            highest = arrF[i];

            pos = i;

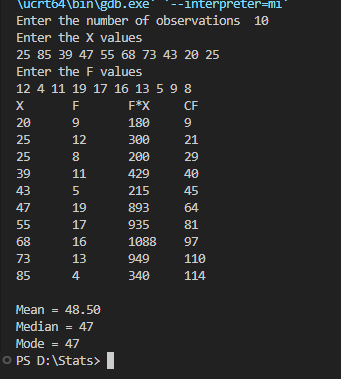
        }

    }

    printf("Mode = %d\n", arrX[pos]);

}

OUTPUT:



Program 3. Write a program to calculate mean, median and mode in a continuous series.

#include<stdio.h>

void main(){

    int size;

    printf("Enter the number of observations  ");

    scanf("%d", &size);

    int upperB[size];

    int lowerB[size];

    printf("Enter the upper and lower bound values\n");

    for(int i=0; i<size; i++){

        scanf("%d%d", &lowerB[i], &upperB[i]);

    }

    int arrF[size];

    printf("Enter the F values\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrF[i]);

    }

    // Making the table

    int mid\_value[size];

    int FM[size];

    int CF[size];

    int add = 0;

    for(int i=0; i< size; i++){

        mid\_value[i] = (upperB[i] + lowerB[i]) / 2;

        FM[i] = mid\_value[i] \* arrF[i];

        add += arrF[i];

        CF[i] = add;

    }

    // Printing the table

    printf("intervals\tFreq\tmid values(M)\tF\*M\tCF\n");

    for(int i=0; i<size; i++){

        printf("%d-%d\t\t%d\t\t%d\t%d\t%d\n",lowerB[i], upperB[i], arrF[i], mid\_value[i], FM[i], CF[i]);

    }

    // //  MEAN

    float sumFX = 0;

    float sumF = 0;

    for(int i=0; i<size; i++){

        sumFX += FM[i];

        sumF += arrF[i];

    }

    float mean = sumFX / sumF;

    printf("\nMean = %.2f\n", mean);

    // MEDIAN

    float term = sumF / 2;

    int positionCF = 0;

    int positionF = 0;

    for(int i=0; i<size; i++){

        if(CF[i] < term){

            int a;

        }else{

            positionF = i;

            positionCF = i - 1;

            break;

        }

    }

    float x = (term - CF[positionCF])/ arrF[positionF];

    float y = (upperB[positionF] - lowerB[positionF]);

    float z = lowerB[positionF];

    printf("Median = %.2f\n", x \* y + z);

    // // MODE

    int highest = 0;

    int f1, f2 , f0 = 0;

    for(int i=0; i< size; i++){

        if(arrF[i] > highest) {

            highest = arrF[i];

            f1 = i;

        }

    }

    if(arrF[f1] == (size - 1)){

        f0 = f1 - 1;

        f2 = 0;

    }else if(arrF[f1] == 0){

        f0 = 0;

        f2 = f1 + 1;

    }else{

        f0 = f1 -1;

        f2 = f1 + 1;

    }

    float a = upperB[positionF] - lowerB[positionF];

    float b = ((2 \* arrF[f1]) - arrF[f0] - arrF[f2]);

    float c = (arrF[f1] - arrF[f0]);

    float d = lowerB[f1];

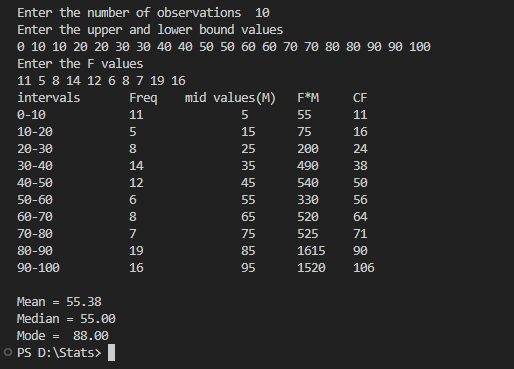
    float a1 =  c/ b;

    float a2 = a1 \* a;

    printf("Mode =  %.2f", a2 + d);

}

OUTPUT:



Program 4. Write a program to calculate geometric and harmonic mean in a discrete series.

#include<stdio.h>

#include<math.h>

void main(){

    int size;

    printf("ENTER the number of elements\n");

    scanf("%d", &size);

    // The X column

    int arrX[size];

    printf("ENTER X elements\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrX[i]);

    }

    // THe F Cloumn

    int arrF[size];

    printf("ENTER the Frequency\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrF[i]);

    }

    // Taking log values

    double logg[size];

    for(int i=0; i<size; i++){

        logg[i] = log(arrX[i]);

    }

    // Taking submission of F column

    int freq\_sum = 0;

    for(int i=0; i<size; i++){

        freq\_sum += arrF[i];

    }

    // Making f/x column

    double arrFX[size];

    for(int i=0; i<size; i++){

        arrFX[i] = (double)arrF[i] / (double)arrX[i];

    }

    // Printing the table

    printf("\n X \t F \t log(x) \t f/x\n");

    for(int i=0; i<size; i++){

        printf("%d \t %d \t %.4lf \t %.4lf\n", arrX[i], arrF[i], logg[i], arrFX[i]);

    }

    //  ------------------------------------ Calculating G.M ------------------------------------

    // Taking Log array sum

    double log\_sum = 0;

    for(int i=0; i<size; i++){

        log\_sum += logg[i];

    }

    // Taking antilog

    double result\_gm  = exp(log\_sum / size);

    printf("Geometric Mean = %.3lf\n", result\_gm);

    // ------------------------------------ Calculating H.M ------------------------------------

    // Taking f/x array sum

    double fx\_sum = 0;

    for(int i=0; i<size; i++){

        fx\_sum += arrFX[i];

    }

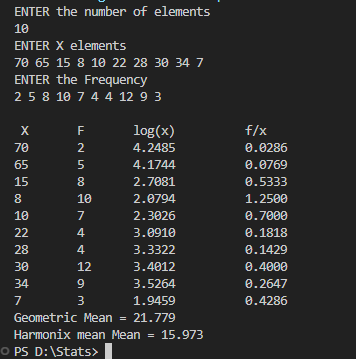
    // Calculating sum

    double result\_hm = freq\_sum / fx\_sum;

    printf("Harmonix mean Mean = %.3lf", result\_hm);

}

OUTPUT:



Program 5. Write a program to calculate mean deviation from mean, median and mode in continuous series.

#include<stdio.h>

#include <math.h>

void main(){

    int size;

    printf("Enter the number of elements\n");

    scanf("%d", &size);

    int upper[size], lower[size];

    printf("Enter the upper and lower bound values\n");

    for(int i=0; i<size; i++){

        scanf("%d%d", &lower[i], &upper[i]);

    }

    int arrF[size];

    printf("Enter the frequency values\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrF[i]);

    }

    // Mid value

    int arrX[size];

    for(int i=0; i<size; i++){

        arrX[i] = (upper[i] + lower[i]) /  2;

    }

    // cf

    int arrCF[size];

    int temp\_sum= 0;

    for(int i=0; i<size; i++){

        temp\_sum += arrF[i];

        arrCF[i] = temp\_sum;

    }

    // fx

    int arrFX[size];

    for(int i=0; i<size; i++){

        arrFX[i] = arrX[i] \* arrF[i];

    }

    // Mean

    int  sumFX=0, sumF=0;

    for(int i=0; i<size; i++){

        sumF += arrF[i];

        sumFX += arrFX[i];

    }

    float mean = (float)sumFX / (float) sumF;

    // Median

    float term = sumF / 2;

    int positionCF = 0;

    int positionF = 0;

    for(int i=0; i<size; i++){

        if(arrCF[i] < term){

            int a;

        }else{

            positionF = i;

            positionCF = i - 1;

            break;

        }

    }

    float x = (term - arrCF[positionCF])/ arrF[positionF];

    float y = (upper[positionF] - lower[positionF]);

    float z = lower[positionF];

    float median =  x \* y + z;

    // // MODE

    int highest = 0;

    int f1, f2 , f0 = 0;

    for(int i=0; i< size; i++){

        if(arrF[i] > highest) {

            highest = arrF[i];

            f1 = i;

        }

    }

    if(arrF[f1] == (size - 1)){

        f0 = f1 - 1;

        f2 = 0;

    }else if(arrF[f1] == 0){

        f0 = 0;

        f2 = f1 + 1;

    }else{

        f0 = f1 -1;

        f2 = f1 + 1;

    }

    float a = upper[positionF] - lower[positionF];

    float b = ((2 \* arrF[f1]) - arrF[f0] - arrF[f2]);

    float c = (arrF[f1] - arrF[f0]);

    float d = lower[f1];

    float a1 =  c/ b;

    float a2 = a1 \* a;

    float mode = a2 + d;

    //  |x - mean|

    float xbar[size];

    for(int i=0; i<size; i++){

        xbar[i] = fabs((float)arrX[i] - mean);

    }

    // f \* |x - mean|

    float xbarf[size];

    for(int i=0; i<size; i++){

        xbarf[i] = (float)xbar[i] \* (float)arrF[i];

    }

    // |x - median|

    float median\_xbar[size];

    for(int i=0; i<size; i++){

        median\_xbar[i] = fabs((float)arrX[i] - median);

    }

    // f \* |x - median|

    float median\_xbarf[size];

    for(int i=0; i<size; i++){

        median\_xbarf[i] = (float)median\_xbar[i] \* (float)arrF[i];

    }

    // |x - mode|

    float mode\_xbar[size];

    for(int i=0; i<size; i++){

        mode\_xbar[i] = fabs((float)arrX[i] - mode);

    }

    // f \* |x - mode|

    float mode\_xbarf[size];

    for(int i=0; i<size; i++){

        mode\_xbarf[i] = (float)mode\_xbar[i] \* (float)arrF[i];

    }

    // Print Table

    printf("Interval\tfreq\tX\tCF\tf\*x\t|x-mean|\tf\*|x-mean|\t|x-median|\tf\*|x-median|\t|x-mode|\tf\*|x-mode|\n");

    for(int i=0; i<size; i++){

        printf("%d-%d\t\t%d\t%d\t%d\t%d\t%.2f\t\t%.2f\t\t%.2f\t\t%.2f\t\t%.2f\t\t%.2f\n",lower[i], upper[i], arrF[i], arrX[i], arrCF[i], arrFX[i], xbar[i], xbarf[i], median\_xbar[i], median\_xbarf[i], mode\_xbar[i], mode\_xbarf[i]);

    }

    // Meadian Deviation from Mean

    float MEAN;

    for(int i=0;i< size; i++){

        MEAN += xbarf[i]; }

    printf("\nMean deviation from MEAN is %.3f", MEAN/ (float)sumF);

    // Meadian Deviation from Median

    float MEDAIN;

    for(int i=0;i< size; i++){

        MEDAIN += median\_xbarf[i];}

    printf("\nMean deviation from MEDAIN is %.3f", MEDAIN/ (float)sumF);

    // Meadian Deviation from Mean

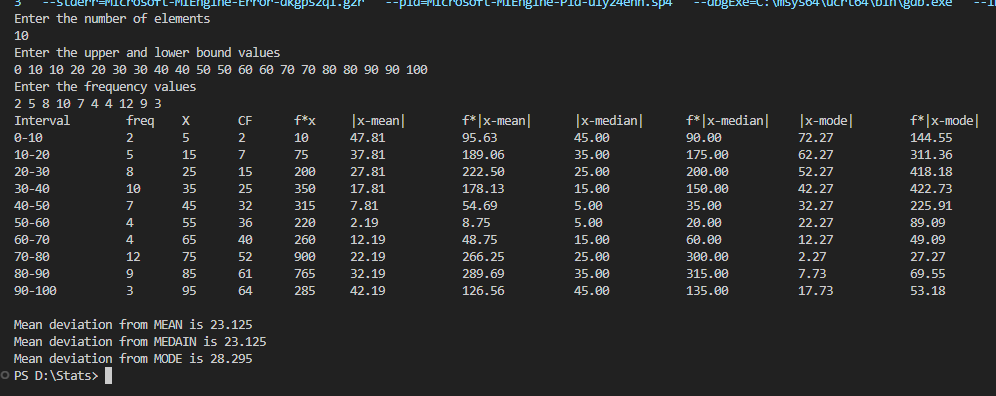
    float MODE;

    for(int i=0;i< size; i++){

        MODE += mode\_xbarf[i];  }

    printf("\nMean deviation from MODE is %.3f", MODE/ (float)sumF);}

OUTPUT:



Program 6. Write a program to calculate standard deviation for continuous series using array.

#include<stdio.h>

#include <math.h>

void main(){

    int size;

    printf("Enter the number of elements\n");

    scanf("%d", &size);

    int upper[size], lower[size];

    printf("Enter the upper and lower bound values\n");

    for(int i=0; i<size; i++){

        scanf("%d%d", &lower[i], &upper[i]);

    }

    int arrF[size];

    printf("Enter the frequency values\n");

    for(int i=0; i<size; i++){

        scanf("%d", &arrF[i]);

    }

    // Mid value

    int arrX[size];

    for(int i=0; i<size; i++){

        arrX[i] = (upper[i] + lower[i]) /  2;

    }

    // fx

    int arrFX[size];

    for(int i=0; i<size; i++){

        arrFX[i] = arrX[i] \* arrF[i];

    }

    // Mean

    int  sumFX=0, sumF=0;

    for(int i=0; i<size; i++){

        sumF += arrF[i];

        sumFX += arrFX[i];

    }

    float mean = (float)sumFX / (float) sumF;

     //  x - mean

    float xbar[size];

    for(int i=0; i<size; i++){

        xbar[i] = (float)arrX[i] - mean;

    }

    //  x - mean ^ 2

    float square\_xbar[size];

    for(int i=0; i<size; i++){

        square\_xbar[i] = xbar[i] \* xbar[i];

    }

    // f \* (x - mean)^2

    float xbarf[size];

    for(int i=0; i<size; i++){

        xbarf[i] = (float)square\_xbar[i] \* (float)arrF[i];

    }

    // Print Table

    printf("Interval\tfreq\tX\tf\*x\t(x-mean)\t(x-mean)^2\tf\*(x-mean)^2\n");

    for(int i=0; i<size; i++){

        printf("%d-%d\t\t%d\t%d\t%d\t%.2f\t\t%.2f\t\t%.2f\n",lower[i], upper[i], arrF[i], arrX[i], arrFX[i], xbar[i], square\_xbar[i], xbarf[i]);

    }

    // Standard Deviation

    float sum\_xbarf;

    for(int i=0; i< size; i++){

        sum\_xbarf += xbarf[i];

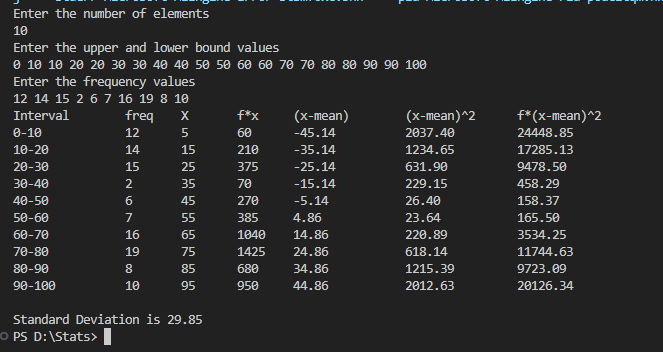
    }

    float SD = sqrt(sum\_xbarf / sumF);

    printf("\nStandard Deviation is %.2f", SD);

}

OUTPUT:



Program 7. Write a program to calculate one period ahead forecast using Naïve method.

#include<stdio.h>

void main(){

    int size;

    printf("Enter the size of the data: ");

    scanf("%d", &size);

    int year[size];

    int  actual\_sales[size];

    printf("Enter the years:\n");

    for(int i=0; i<size; i++){

        scanf("%d", &year[i]);

    }

    printf("Enter the actual sales:\n");

    for(int i=0; i<size; i++){

        scanf("%d", &actual\_sales[i]);

    }

    actual\_sales[size-1] = -1;

    int forecast\_by\_nave\_method[size];

    forecast\_by\_nave\_method[0] = 0;

    for(int i=1; i<size; i++){

        forecast\_by\_nave\_method[i] = actual\_sales[i-1];

    }

    // Printing the table

    printf("Year\tAcutal Sales\tForecast By Nave Method\n");

    for(int i=0; i<size; i++){

        if (actual\_sales[i] == -1){

            printf("%d\t?\t\t%d\n", year[i], forecast\_by\_nave\_method[i]);

        }else{

            printf("%d\t%d\t\t%d\n", year[i], actual\_sales[i], forecast\_by\_nave\_method[i]);

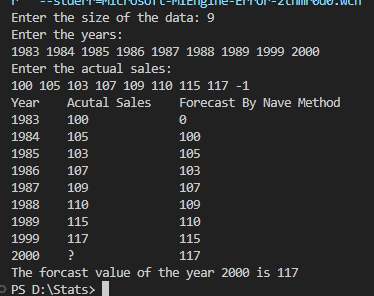
        }

    }

    printf("The forcast value of the year %d is %d", year[size-1], forecast\_by\_nave\_method[size-1]);

}

OUTPUT:



Program 8. Write a program to calculate one period ahead forecast using moving average method.

#include<stdio.h>

void main(){

    int size;

    printf("Enter the size of the data: ");

    scanf("%d", &size);

    int year[size];

    int  actual\_sales[size];

    printf("Enter the years:\n");

    for(int i=0; i<size; i++){

        scanf("%d", &year[i]);

    }

    printf("Enter the actual sales:\n");

    for(int i=0; i<size; i++){

        scanf("%d", &actual\_sales[i]);

    }

    actual\_sales[size-1] = -1;

    int forecast\_by\_nave\_method[size];

    forecast\_by\_nave\_method[0] = 0;

    forecast\_by\_nave\_method[1] = 0;

    forecast\_by\_nave\_method[2] = 0;

    for(int i=3; i<size; i++){

        forecast\_by\_nave\_method[i] = actual\_sales[i-1] + actual\_sales[i-2] + actual\_sales[i-3];

    }

    // Printing the table

    printf("Year\tAcutal Sales\tForecast By Nave Method\n");

    for(int i=0; i<size; i++){

        if (actual\_sales[i] == -1){

            printf("%d\t?\t\t%d\n", year[i], forecast\_by\_nave\_method[i]);

        }else{

            printf("%d\t%d\t\t%d\n", year[i], actual\_sales[i], forecast\_by\_nave\_method[i]);

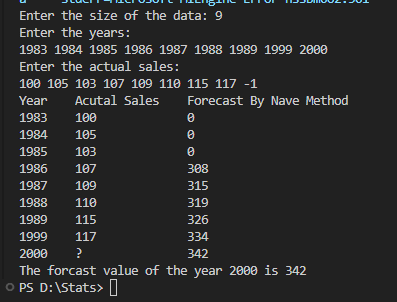
        }

    }

    printf("The forcast value of the year %d is %d", year[size-1], forecast\_by\_nave\_method[size-1]);

}

OUTPUT:



Program 9. Write a program to calculate standard statistical measures using exponential smoothing forecasting method. [Mean Error (ME), Mean Absolute Error (MAE), Mean Square Error (MSE)]

#include<stdio.h>

#include<math.h>

void main(){

    int size;

    printf("Enter the size of the duration: ");

    scanf("%d", &size);

    int demand[size];

    printf("Enter the demand:\n");

    for(int i=0; i<size; i++){

        scanf("%d", &demand[i]);

    }

    float alpha;

    printf("Enter the alpha value:\n");

    scanf("%f", &alpha);

    float intial;

    printf("Enter the intial value of the forcaste:\n");

    scanf("%f", &intial);

    // Forcaste table

    float forcaste[size];

    forcaste[0]= intial;

    for(int i=1; i<size; i++){

        forcaste[i] =  forcaste[i-1] + ((float)alpha \* (float)(demand[i-1] - forcaste[i-1]));

    }

    // Mean Error Table

    float me[size], mean\_error = 0;

    for(int i=0; i<size; i++){

        me[i] = demand[i] - forcaste[i];

        mean\_error += demand[i] - forcaste[i];

    }

    // Mean Square Error Table

    float mse[size], mean\_square\_error = 0;

    for(int i=0; i<size; i++){

        mse[i] = me[i] \* me[i];

        mean\_square\_error += (me[i] \* me[i]);

    }

    // Mean Absolute Error Table

    float mae[size], mean\_absolute\_error;

    for(int i=0; i<size; i++){

        mae[i] = fabs(demand[i] - forcaste[i]);

        mean\_absolute\_error += fabs(demand[i] - forcaste[i]);

    }

    // Print table

    printf("Duration\tDemand\tForcaste\tMeanError\tMeanSquareError\t  MeanAbsoluteError\n");

    for (int i = 0; i < size; i++){

        printf("%d\t\t%d\t%f\t%f\t%f\t  %f\n", i+1, demand[i], forcaste[i], me[i], mse[i], mae[i]);

    }

    float pred = forcaste[size-1] + alpha \* (demand[size-1] - forcaste[size-1]);

    printf("%d\t\t0\t%f", size+1, pred);

    printf("\n");

    // Mean Errors

    printf("Mean Error = %.2f\n", mean\_error / size);

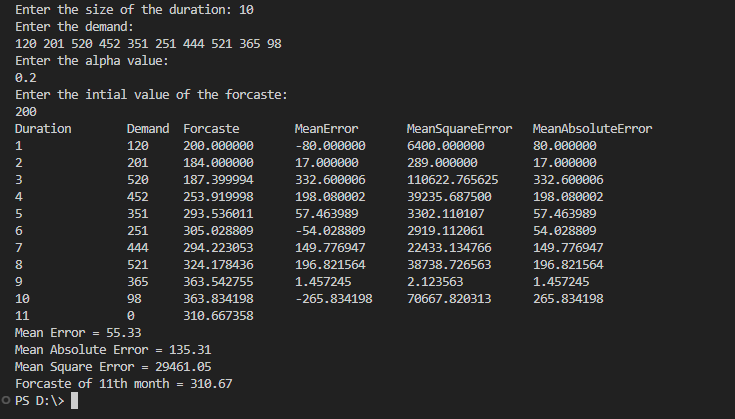
    printf("Mean Absolute Error = %.2f\n", mean\_absolute\_error / size);

    printf("Mean Square Error = %.2f\n", mean\_square\_error / size);

    printf("Forcaste of 11th month = %.2f", pred);

}

OUTPUT:



Program 10. Write a program to calculate relative measures of forecasting using exponential smoothing forecasting method. [Percentage Error (PE), Mean Percentage Error (MPE), mean Absolute Percentage Error (MAPE)]

#include<stdio.h>

#include<math.h>

void main(){

    int size;

    printf("Enter the size of the duration: ");

    scanf("%d", &size);

    int demand[size];

    printf("Enter the demand:\n");

    for(int i=0; i<size; i++){

        scanf("%d", &demand[i]);

    }

    float alpha;

    printf("Enter the alpha value:\n");

    scanf("%f", &alpha);

    float intial;

    printf("Enter the intial value of the forcaste:\n");

    scanf("%f", &intial);

    // Forcaste table

    float forcaste[size];

    forcaste[0]= intial;

    for(int i=1; i<size; i++){

        forcaste[i] =  forcaste[i-1] + ((float)alpha \* (float)(demand[i-1] - forcaste[i-1]));

    }

    // Percent Error Table

    float pe[size];

    for(int i=0; i<size; i++){

        pe[i] = (float)((demand[i] - forcaste[i]) / (float)demand[i]) \* 100.00;

    }

    // Mean percent Table

    float mpe[size], mean\_percent\_error = 0;

    for(int i=0; i<size; i++){

        mpe[i] = pe[i];

        mean\_percent\_error += pe[i] ;

    }

    // Mean Absolute Percent Error Table

    float mape[size], mean\_percent\_absolute\_error;

    for(int i=0; i<size; i++){

        mape[i] = fabs(pe[i]);

        mean\_percent\_absolute\_error += fabs(pe[i]);

    }

    // Print table

    printf("Duration\tDemand\tForcaste\tPercentError\t  MeanPercentAbsoluteError\n");

    for (int i = 0; i < size; i++){

        printf("%d\t\t%d\t%f\t%f\t \t %f\n", i+1, demand[i], forcaste[i], pe[i], mape[i]);

    }

    float pred = forcaste[size-1] + alpha \* (demand[size-1] - forcaste[size-1]);

    printf("%d\t\t0\t%f", size+1, pred);

    printf("\n");

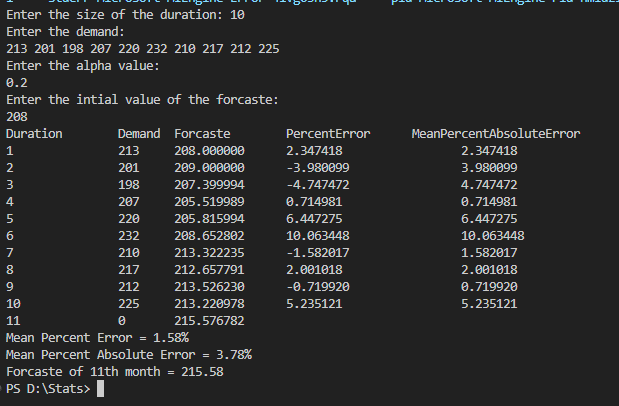
    // Mean Errors

    printf("Mean Percent Error = %.2f%%\n", mean\_percent\_error / size);

    printf("Mean Percent Absolute Error = %.2f%%\n", mean\_percent\_absolute\_error / size);

    printf("Forcaste of 11th month = %.2f", pred);

}

 OUTPUT:

Program 11. Write a program to generate random numbers using mid square method.

#include<stdio.h>

void main(){

   int seedNum,n1,n2,n3,n;

   printf("Enter the 4 digit seed number : ");

   scanf("%d",&seedNum);

   printf("Enter count to genrate random numbers : ");

   scanf("%d",&n);

   printf("Generated random numbers :\n%d\n",seedNum);

   for(int i=0; i<n ; i++){

      n1=seedNum\*seedNum;

      n2=n1/100;

      n3=n2%10000;

      printf("%d\n",n3);

      seedNum=n3;

      }

}

OUTPUT:

