//Chonghao Gu HW#7B

//CMPSC 201 Spring 2022

//Date 2022/3/31

//grades

#include <iostream>

using namespace std;

double average(double x[], int n);

double maximum(double x[], int n);

double minimum(double x[], int n);

int nAboveAvg(double x[], int n);

void sort(double x[], int n);

int main()

{

double grades[50];

int ngrades;

cout<<"How many grades? (max = 50) ";

cin>>ngrades;

for(int i = 0; i<ngrades; i++)

{

cout<<"Enter grade ";

cin>> grades[i];

while(grades[i]< 0 || grades[i] > 100)

{

cout<<"Invalid grade- please enter again"<<endl;

cin>>grades[i];

if(grades[i]>= 0 && grades[i] <= 100)

{

break;

}

}

}

double avg = average(grades, ngrades);

double max = maximum(grades, ngrades);

double min = minimum(grades, ngrades);

int nAbove = nAboveAvg(grades, ngrades);

cout << "Average = " << avg << endl;

cout << "# above average = " << nAbove << endl;

cout<<"Max value is = "<<max<<endl;

cout<<"Min value is = "<<min<<endl;

sort(grades,ngrades);

cout<<endl<<"Grades in Sorted Ascending Order: "<<endl;

for(int i=0;i<ngrades;i++)

{

cout<<" "<<grades[i];

}

}

double average(double x[], int npts)

{

double sum = 0;

for(int k = 0; k<npts; k++)

{

sum = sum +x[k];

}

return sum / npts;

}

double maximum(double x[], int npts)

{

double max = x[0];

for(int i = 0; i < npts; i++)

{

if(x[i] > max)

{

max = x[i];

}

}

return max;

}

double minimum(double x[], int npts)

{

double min = x[0];

for(int i = 0; i < npts; i++)

{

if(x[i] < min)

{

min = x[i];

}

}

return min;

}

int nAboveAvg(double x[], int npts)

{

int count = 0;

double avg = average(x, npts);

for(int i=0; i < npts; ++i)

{

if(x[i] > avg)

{

count++;

}

}

return count;

}

void sort(double x[], int npts)

{

double min\_value;

int min\_index;

double temp;

for (int i=0; i < npts-1; i++)

{

min\_value = x[i];

min\_index = i;

for (int j=i+1; j< npts; j++)

{

if (x[j] < min\_value)

{

min\_value = x[j];

min\_index = j;

}

}

temp = x[min\_index];

x[min\_index] = x[i];

x[i] = temp;

}

return;

}

Test：





