#include<iostream>

using namespace std;

void displayArray(int array[], int size);

void fillArray(int array[], int size);

void selectionSort(int array[], int size);

int findMax(int array[], int size);

int findMin(int array[], int size);

double findAvg(int array[], int size);

int main()

{

const int maxSize = 50;

int scores[maxSize];

int count;

cout << "How many scores do you want to enter?: ";

cin >> count;

cout << endl;

if (count > 0 && count <= 50)

{

fillArray(scores, count);

cout << "Original order scores are: ";

displayArray(scores, count);

selectionSort(scores, count);

cout << endl<< "Scores sorted high to low: ";

displayArray(scores, count);

cout <<endl<< "Highest score: " << findMax(scores, count) << endl

<< "Lowest score: " << findMin(scores, count) << endl

<< "Avarage score: " << findAvg(scores, count)<<endl<<endl;

return 0;

}

else

{

cout << "Invalid input please try again!!"<< endl<<endl;

}

system("pause");

return 0;

}

void fillArray(int array[], int size)

{

for (int i = 0, scoreNum = 1; i < size; i++,scoreNum++)

{

cout << "Enter score #" << scoreNum << " : ";

cin >> array[i];

}

}

void displayArray(int array[], int size)

{

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

{

cout << array[i] << ", ";

}

}

void selectionSort(int array[], int size)

{

int maxIndex, maxValue;

for (int start = 0; start < (size - 1); start++)

{

maxIndex = start;

maxValue = array[start];

for (int index = start + 1; index < size; index++)

{

if (array[index] > maxValue)

{

maxValue = array[index];

maxIndex = index;

}

}

swap(array[maxIndex], array[start]);

}

}

double findAvg(int array[], int size)

{

int sum = 0;

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

{

sum += array[i];

}

return ((double)sum / size);

}

int findMin(int array[], int size)

{

int min = array[0];

for (int i = 1; i < size; i++)

{

if (array[i] < min)

min = array[i];

}

return min;

}

int findMax(int array[], int size)

{

int max = array[0];

for (int i = 1; i < size; i++)

{

if (array[i] > max)

max = array[i];

}

return max;

}









