```
/*
point of this is to demonstrate some array manipulations required for running average or sometimes called Moving Average
(MA filter).
There are also some extra functions for standard deviation and variance.

All functions work with any size of an array. The size of the array just needs to be defined at startup, although
it can change throughout the program. example: if you want 3 items in the array put double ArrayTest[] = {0,0,0};
for ten: double ArrayTest[] = {0,0,0,0,0,0,0,0,0,0};

*/

double ArrayTest[] = {0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,0};

double sampleLoop = 0;//temp

void setup() {

 Serial.begin(115200);

}

void loop() {


    storeValue(sampleLoop);
    arrayOutPut();


    Serial.print("array average:");
    Serial.println(averageArray(),7);

    Serial.print("Standard Deviaion:");
    Serial.println(standardDeviation(),7);
    Serial.print("variance:");
    Serial.println(variance(),7);

    Serial.print("\n");
    delay(3000);


   sampleLoop++;



}

void arrayOutPut(){
  int arrayIndex = (sizeof ArrayTest/sizeof(int))/2; //find the number of entries in array

  Serial.print("\n");

  Serial.print("array size:");
  Serial.println(arrayIndex);

  for(int i = 0; i <= (arrayIndex-1); i++){//since for loop..
    Serial.print("value for:");
    Serial.print(i);
    Serial.print(" array value:");
    Serial.println(ArrayTest[i]);
  }

 // Serial.print("\n");

}

void storeValue(double newValue){ //shifts all the values down and the value sent goes into the 0'th element. last
element deleted

  int arrayIndex = (sizeof ArrayTest/sizeof(int))/2; //find the number of entries in array

  for(int i = (arrayIndex-1); i >= 0 ; i--){ //-1 since for loop..
     ArrayTest[i] = ArrayTest[i-1];
  }

  ArrayTest[0] = newValue;
  return;
}


double averageArray(){
```

```
  int arrayIndex = (sizeof ArrayTest/sizeof(int))/2; //find the number of entries in array
  double sum = 0;

  for(int i = 0; i <= (arrayIndex-1); i++){ //-1 since for loop..
      sum += ArrayTest[i];
  }

  return (sum/arrayIndex);
}


double standardDeviation(){

  int arrayIndex = (sizeof ArrayTest/sizeof(int))/2; //find the number of entries in array
  double sumXsquared = 0;
  double sumX = 0;

  //sum up the individual squared values
  for(int i = 0; i <= (arrayIndex-1); i++){ //-1 since for loop..
      sumXsquared += pow(ArrayTest[i],2);
  }

  //sum up the values then square sumX
  for(int i = 0; i <= (arrayIndex-1); i++){ //-1 since for loop..
      sumX += ArrayTest[i];
  }
  sumX = pow(sumX,2);

  return (sumXsquared - sumX/((double)arrayIndex))/((double)arrayIndex - 1.);

}

double variance(){

  //this is just the square root of standard deviation.
  //you could just have this method square root the S.D. output
  //return pow(standardDeviation(),.5);


  int arrayIndex = (sizeof ArrayTest/sizeof(int))/2; //find the number of entries in array
  double sumXsquared = 0;
  double sumX = 0;

  //sum up the individual squared values
  for(int i = 0; i <= (arrayIndex-1); i++){ //-1 since for loop..
      sumXsquared += pow(ArrayTest[i],2);
  }

  //sum up the values then square sumX
  for(int i = 0; i <= (arrayIndex-1); i++){ //-1 since for loop..
      sumX += ArrayTest[i];
  }
  sumX = pow(sumX,2);

  return pow((sumXsquared - sumX/((double)arrayIndex))/((double)arrayIndex - 1.),.5);

}
```