Tutorial #12: Arrays

Question 1:

You are given the main method. Complete the class by writing the `standardDev` method such that the program prints the standard deviation for the given arrays.

$$\sigma = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(x_i - \mu)^2}$$

```
public class Q1
{
      public static void main(String[] args)
       {
            double[] ex1 = {1.5, 1.5, 1.5};
            double[] ex2 = {0.0, 1.0, 2.0, 3.0};
            double[] ex3 = {3.4, 5.6, 123.4, 684.93, 3.4};

            System.out.println(standardDev(ex1, ex1.length));
            System.out.println(standardDev(ex2, ex2.length));
            System.out.println(standardDev(ex3, ex3.length));
       }

      // definition of standardDev goes here

}
```

Question 2:

Consider this code fragment. What geometric figure best describes the array that it constructs? Can you make a picture showing its contents?

```
 char ch = 'A';
 char[][] ary = new char[5][];

 for (int k = 0; k < ary.length; k++)
 {
     ary[k] = new char[k+1];
     for (int j = 0; j < ary[k].length; j++)
     ary[k][j] = ch++;
 }
```

Question 3:

In each of these code fragments, predict what is printed and then run the programs to check your answers.

A.

```
public class Q3_Array_3A
{
   public Q3_Array_3A()
   {
      char[] a = { 'a', 'b', 'c', 'd', 'e' };
      change(a);
      System.out.println(new String(a));
   }

   private void change(char[] a)
   {
      for (int i=0; i<a.length/2; i++)
      {
         char c = a[i];
         a[i] = a[a.length - i - 1];
         a[a.length - i - 1] = c;
      }
   }
}


public class Driver3A
{
   public static void main(String[] args) {
      new Q3_Array_3A();
   }
}
```

B.

```
public class Q3_Array_3B
{
   public Q3_Array_3B()
   {
      int[] a = create(2047461578);
      System.out.println(evs(a));
   }

   private int evs(int[] a)
   {
      int n = 0;
      for (int i = 0; i < a.length; i++)
         if (a[i] % 2 == 0)
            ++n;
      return n;
   }

   private int[] create(long n)
   {
      long nn = n;
      int   cnt = 0;
      while (nn > 0)
      {
         nn /= 10;
         ++cnt;
      }
      int[] a = new int[cnt];
      for (int i=0; n>0; i++)
      {
         a[i] = (int) (n % 10); n /= 10;
      }
      return a;
   }
}

public class Driver3B {
   public static void main(String[] args)
   {
      new Q3_Array_3B();
   }
}
```

C.

```
public class Q3_Array_3C
{
      public Q3_Array_3C()
      {
         int[] a = create(24746, 5);
         int[] b = create(35879, 5);
         System.out.print(less(a, b) + " ");
         a = create(247461, 3);
         b = create(358790, 3);
         System.out.println(less(b, a));
      }

      private int[] create(int n, int lth)
      {
         int[] a = new int[lth];
         for (int i=0; i<lth; i++)
         {
            a[i] = n % 10;
            n = n / 10;
         }
         return a;
   }

   private boolean less(int[] a, int[] b)
   {
         boolean isLess = true;
         for (int i=0; i<a.length; i++)
            if (a[i] >= b[i])
            {
               isLess = false;
               break;
            }
         return isLess;
      }
}

public class Driver3C {
   public static void main(String[] args)
   {
      new Q3_Array_3C();
   }
}
```

D.

```
public class Q3_Array_3D
{
   public Q3_Array_3D()
   {
      int[] a = create(6);
      arrange(a);
      arrange(a);
      System.out.println(a[0] + " " + a[1] + " " + a[2]);
   }

   int[] create(int n)
   {
      int[] a = new int[n];
      for (int i=0; i<a.length; i += 2)
      {
         a[i]  = i + 1;
         a[i+1] = -a[i];
      }
      return a;
   }

   void arrange(int[] a) {
      for (int i=0; i<a.length-1; i++)
         if (a[i] < a[i+1])
         {
            int t = a[i];
            a[i] = a[i+1];
            a[i+1] = t;
         }
   }

public class Driver3D {
   public static void main(String[] args)
   {
      new Q3_Array_3D();
   }
}
```

E.

```
public class Q3_Array_3E
{
   public Q3_Array_3E()
   {
      int[][] a = create(6); int i = a.length - 1; for (int j=1; j<=i; j++)
         System.out.print(a[i][j] + " "); System.out.println();
   }

   int[][] create(int n)
   {
      int[][] a = new int[n][];
      a[1] = new int[3];
      a[1][1] = 1;
      for (int i=2; i<a.length; i++)
      {
         a[i] = new int[i+2];
         for (int j=1; j<=i; j++)
         {
            a[i][j] = a[i-1][j-1] + a[i-1][j];
         }
      }
      return a;
   }
}

public class Driver3E
{
   public static void main(String[] args)
   {
      new Q3_Array_3E();
   }
}
```

Question 4:

What condition should be added to the highlighted if statement below to make the fragment create the transpose of the 2-D array? A transpose has the property that:

Transpose[**j**][**k**] = **Original**[**k**][**j**]

```
int[][] twod = { { 1, 2, 3 },{ 4, 5, 6 },{ 7, 8, 9 }};

final int LIM = twod.length;
for (int j = 0; j < LIM; j++)
  for (int k = 0; k < LIM; k++)
    if ( /*add code here*/ )
    {
      int x = twod[j][k];
      twod[j][k] = twod[k][j];
      twod[k][j] = x;
    }
```

Question 5:

Which of these methods will sort an array of floats into ascending order?

A.

```
void arrange(float[] ary)
{
  for (int n=0; n<ary.length; n++)
    for (int k=n; k<ary.length; k++)
    if (ary[n] > ary[k])
    {
      float x = ary[n];
      ary[n] = ary[k];
      ary[k] = x;
    }
}
```

B.

```
void arrange(float[] ary)
{
  for (int n=0; n<ary.length; n++)
    for (int k=n; k<ary.length; k++)
      if (ary[n] > ary[k])
        float x = ary[n];
      ary[n] = ary[k];
      ary[k] = x;
}
```

C.

```
void arrange(float[] ary)
{
   for (int n=1; n<=ary.length; n++)
      for (int k=n; k<ary.length; k++)
         if (ary[n] > ary[k])
         {
            float x = ary[n];
            ary[n] = ary[k];
            ary[k] = x;
         }
}
```

D.

```
void arrange(float[] ary)
{
   for (int n=0; n<ary.length; n++)
      for (int k=n; k<ary.length; k++)
      if (ary[n] > ary[k])
      {
         float x = ary[n];
         ary[n] = ary[k];
         ary[k] = x;
      }
}
```