Exam 2

● Graded

Student

AKSHAJ KAMMARI

Total Points

115 / 130 pts

Question 1

Problem 1: Algorithm Implementation and Basic Java **30** / 30 pts

1.1 **The program Sale reads three arguments from the command-line: (1) a String with the description of an item, (2) a double with the price of the item, and (3) an integer discount code that will determine the amount of discount that will be applied to the item price. The discount code is 0, 1, or 2, where 0 means no discount, 1 means 10% discount, and 2 mean 20% discount. The program displays the item description and its final price after the discount is applied.** **15** / 15 pts

✓ **+ 15 pts** Correct

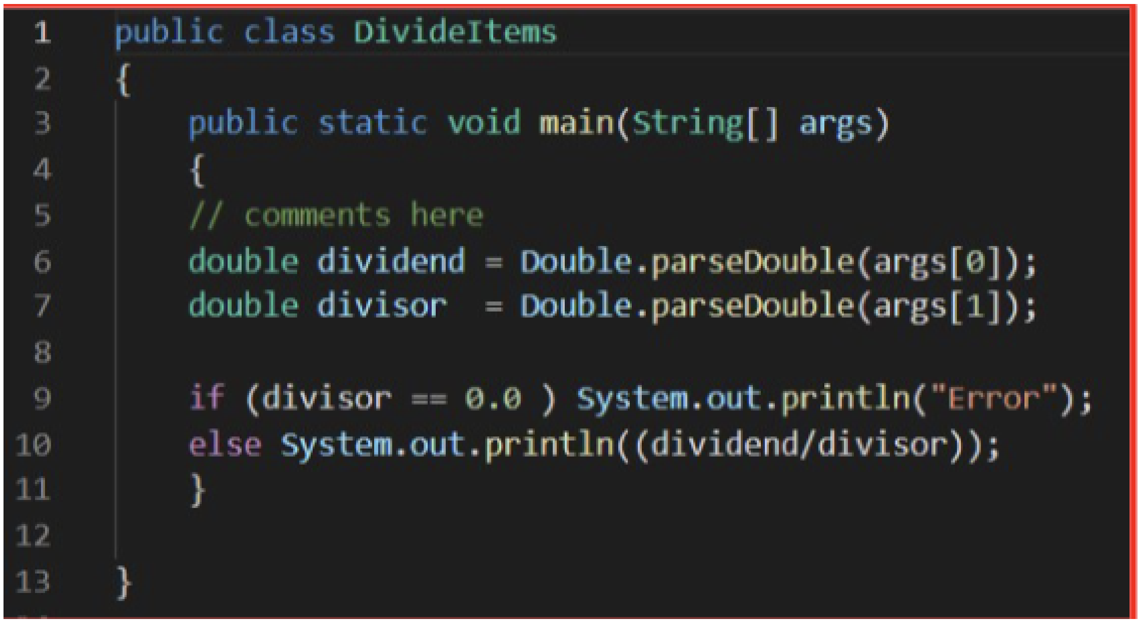
+ 0 pts Incorrect

+ 10 pts partially correct

+ 5 pts partially correct

1.2 **Translate the following pseudocode into a Java program called DivideItems. Assume that dividend and divisors are real numbers. Use the text box or a file upload for your answer.** **15** / 15 pts

✓ **+ 15 pts** correct



```
public class DivideItems
{
    public static void main(String[] args)
    {
    // comments here
    double dividend = Double.parseDouble(args[0]);
    double divisor  = Double.parseDouble(args[1]);

    if (divisor == 0.0 ) System.out.println("Error");
    else System.out.println((dividend/divisor));
    }

}
```

+ 0 pts incorrect

− 3 pts missing declarations

− 3 pts missing semicolons

− 3 pts used int for quotient declaration instead of doubles

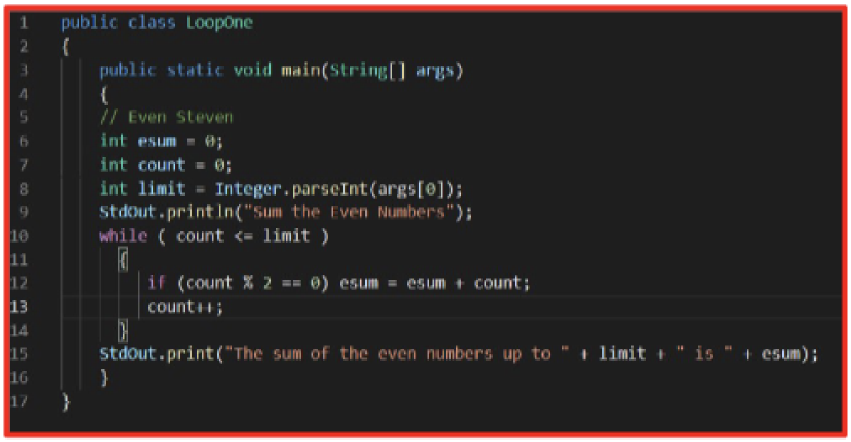
Question 2

Problem 2: Loops **30** / 30 pts

2.1 **The following Java program is intended to take an integer limit from the command line and return the sum of all the even integers between 0 and the limit inclusive (we assume the input is an integer here). The code does not work as intended. Describe the changes to be made in the code so that the correct sum is calculated. Use the text box or a file upload for your answer.** **15** / 15 pts

✔ **+ 15 pts** Correct

Answer: endless loop – need a "count++;" or a "count = count + 1;" after the "if"

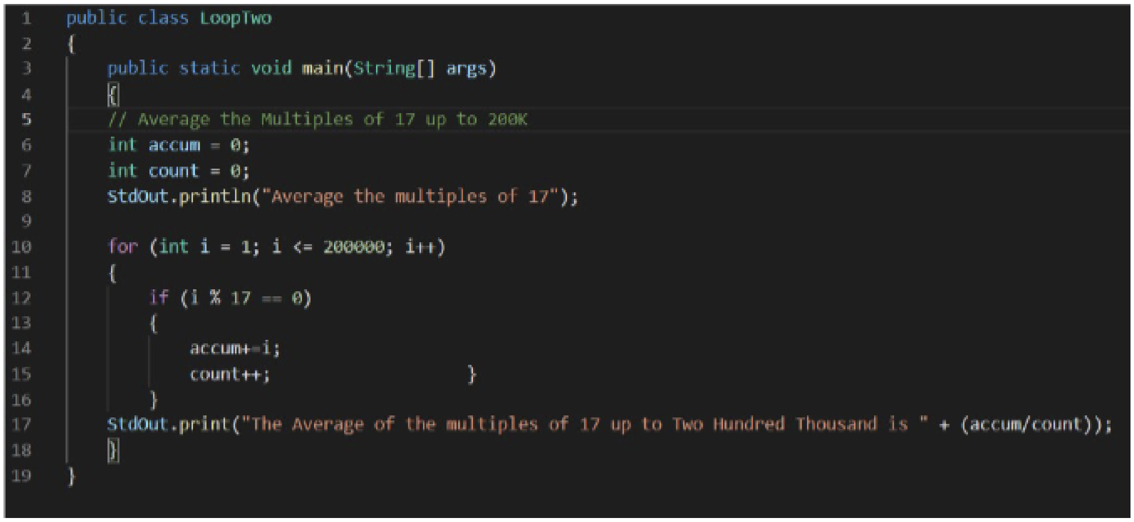


```
public class LoopOne
{
    public static void main(String[] args)
    {
    // Even Steven
    int esum = 0;
    int count = 0;
    int limit = Integer.parseInt(args[0]);
    StdOut.println("Sum the Even Numbers");
    while ( count <= limit )
      {
        if (count % 2 == 0) esum = esum + count;
        count++;
      }
    StdOut.print("The sum of the even numbers up to " + limit + " is " + esum);
    }
}
```

+ 0 pts incorrect

2.2 **Write a Java program that averages the multiples of 17 for the first two hundred thousand natural numbers. Use the text box or a file upload for your answer.** **15** / 15 pts

✔ **+ 15 pts** Correct



```java
public class LoopTwo
{
    public static void main(String[] args)
    {
    // Average the Multiples of 17 up to 200K
    int accum = 0;
    int count = 0;
    StdOut.println("Average the multiples of 17");

    for (int i = 1; i <= 200000; i++)
    {
        if (i % 17 == 0)
        {
            accum+=i;
            count++;                    }
        }
    StdOut.print("The Average of the multiples of 17 up to Two Hundred Thousand is " + (accum/count));
    }
}
```

+ 0 pts incorrect

+ 5 pts loops correctly

+ 5 pts accumulates sum correctly

− 2 pts missing semicolons, declaration of counter, or sum variable

Question 3

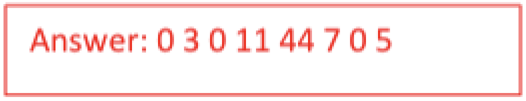
Problem 3: Arrays **25** / 40 pts

3.1 (no title) **10** / 10 pts

+ 10 pts Correct

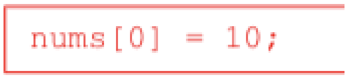
✓ **+ 2 pts** A. double[] a = new double[5];

✓ **+ 2 pts** B.



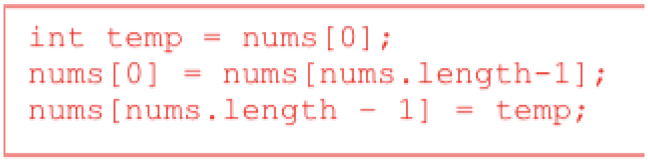
Answer: 0 3 0 11 44 7 0 5

✓ **+ 2 pts** C.



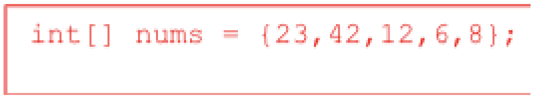
```
nums[0] = 10;
```

✓ **+ 2 pts** D.



```
int temp = nums[0];
nums[0] = nums[nums.length-1];
nums[nums.length - 1] = temp;
```

✓ **+ 2 pts** E.



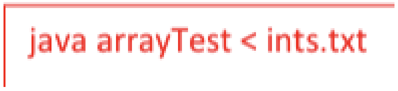
```
int[] nums = {23,42,12,6,8};
```

+ 0 pts incorrect

3.2 The java program ArrayTest.java is intended to do the following: **15** / 15 pts

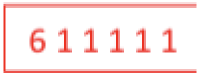
✓ **+ 15 pts** Correct

+ 5 pts A.



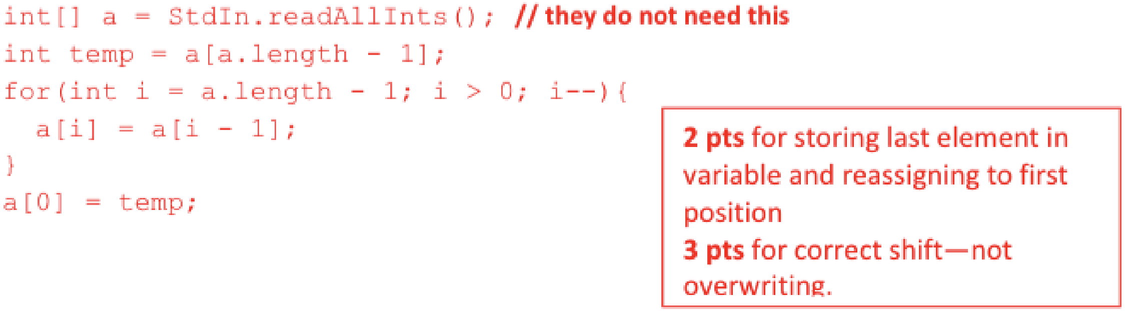
java arrayTest < ints.txt

+ 5 pts B.



6 1 1 1 1 1

+ 5 pts C.



```
int[] a = StdIn.readAllInts();  // they do not need this
int temp = a[a.length - 1];
for(int i = a.length - 1; i > 0; i--){
  a[i] = a[i - 1];
}
a[0] = temp;
```

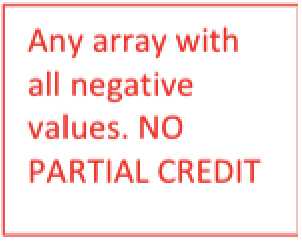
2 pts for storing last element in variable and reassigning to first position
3 pts for correct shift—not overwriting.

+ 0 pts incorrect

Awesome Job :)

3.3 (no title) **0** / 5 pts

+ 5 pts Correct



Any array with all negative values. NO PARTIAL CREDIT

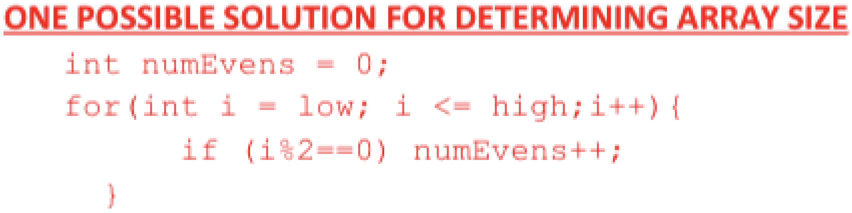
✓ **+ 0 pts** incorrect

3.4 (no title) **0** / 10 pts

+ 10 pts Correct

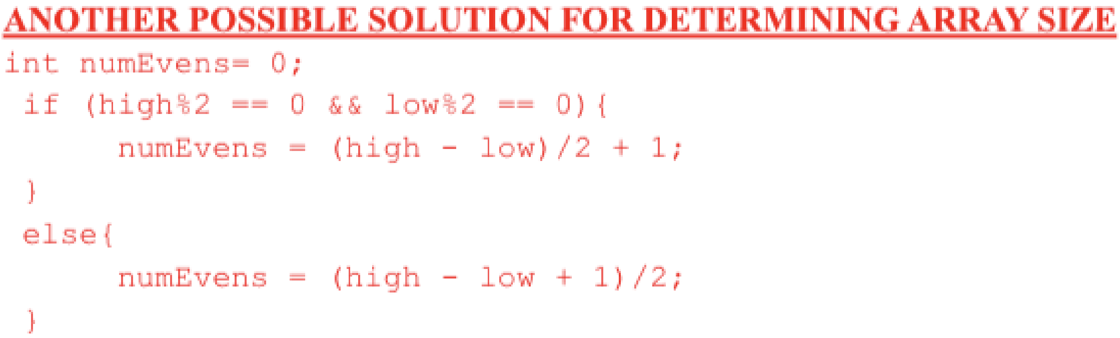
✔ **+ 0 pts** incorrect

+ 5 pts



ONE POSSIBLE SOLUTION FOR DETERMINING ARRAY SIZE

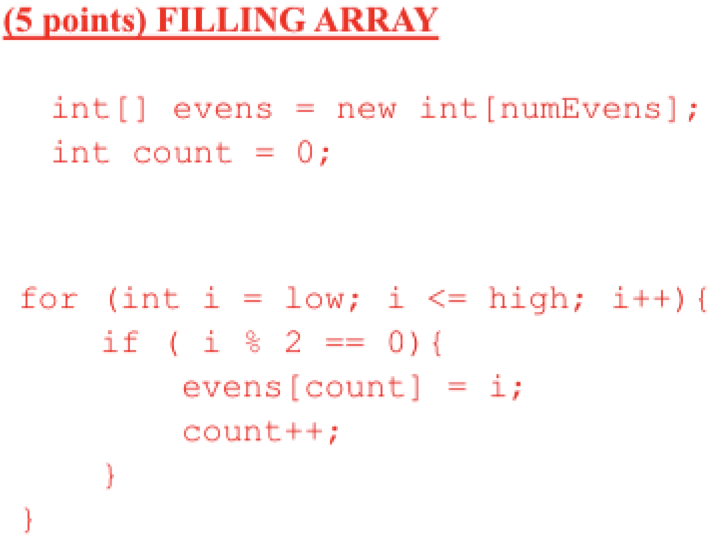
```
int numEvens = 0;
for(int i = low; i <= high;i++){
      if (i%2==0) numEvens++;
  }
```



ANOTHER POSSIBLE SOLUTION FOR DETERMINING ARRAY SIZE

```
int numEvens= 0;
 if (high%2 == 0 && low%2 == 0){
      numEvens = (high - low)/2 + 1;
 }
 else{
      numEvens = (high - low + 1)/2;
 }
```

+ 5 pts



(5 points) FILLING ARRAY

```
int[] evens = new int[numEvens];
int count = 0;


for (int i = low; i <= high; i++){
     if ( i % 2 == 0){
          evens[count] = i;
          count++;
     }
}
```

+ 1 pt partial credit

Assume numEvens has a correct value when grading this part.

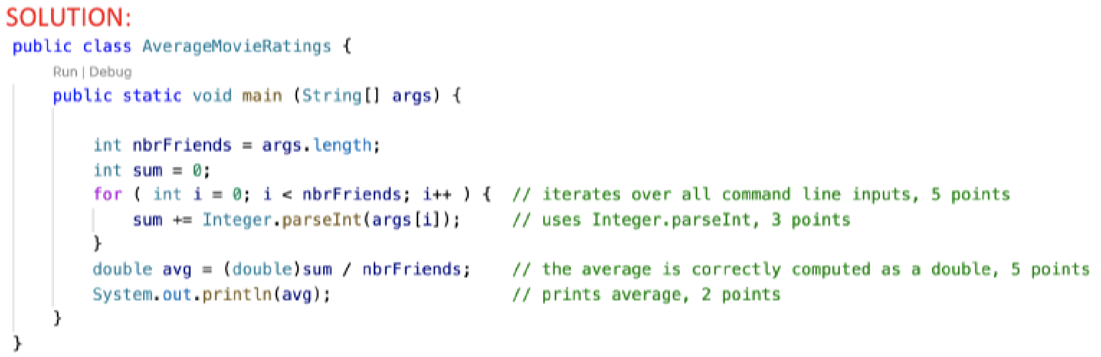
1 pt for creating array
2 pts for putting only even values in the array.
2 pts for putting values in all of the positions in the array….using some variable or way to get from one position to the next.

Question 4

Problem 4: Input and Output **30** / 30 pts

4.1 **(no title)** **15** / 15 pts

✓ **+ 15 pts** Correct



SOLUTION:

```
public class AverageMovieRatings {
    Run | Debug
    public static void main (String[] args) {

        int nbrFriends = args.length;
        int sum = 0;
        for ( int i = 0; i < nbrFriends; i++ ) {  // iterates over all command line inputs, 5 points
            sum += Integer.parseInt(args[i]);     // uses Integer.parseInt, 3 points
        }
        double avg = (double)sum / nbrFriends;    // the average is correctly computed as a double, 5 points
        System.out.println(avg);                  // prints average, 2 points
    }
}
```

+ 5 pts Iterates over command line

+ 3 pts Uses Integer.parseInt

+ 2 pts Prints average

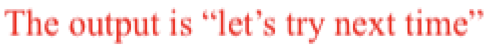
+ 5 pts Average is computed as a double

+ 0 pts Incorrect

4.2 **(no title)** **15** / 15 pts

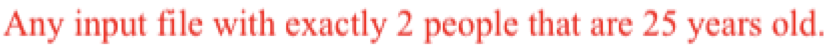
✓ **+ 15 pts** Correct

+ 5 pts A.



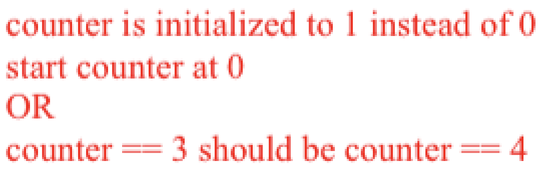
The output is "let's try next time"

+ 5 pts B.



Any input file with exactly 2 people that are 25 years old.

+ 5 pts C.



counter is initialized to 1 instead of 0
start counter at 0
OR
counter == 3 should be counter == 4

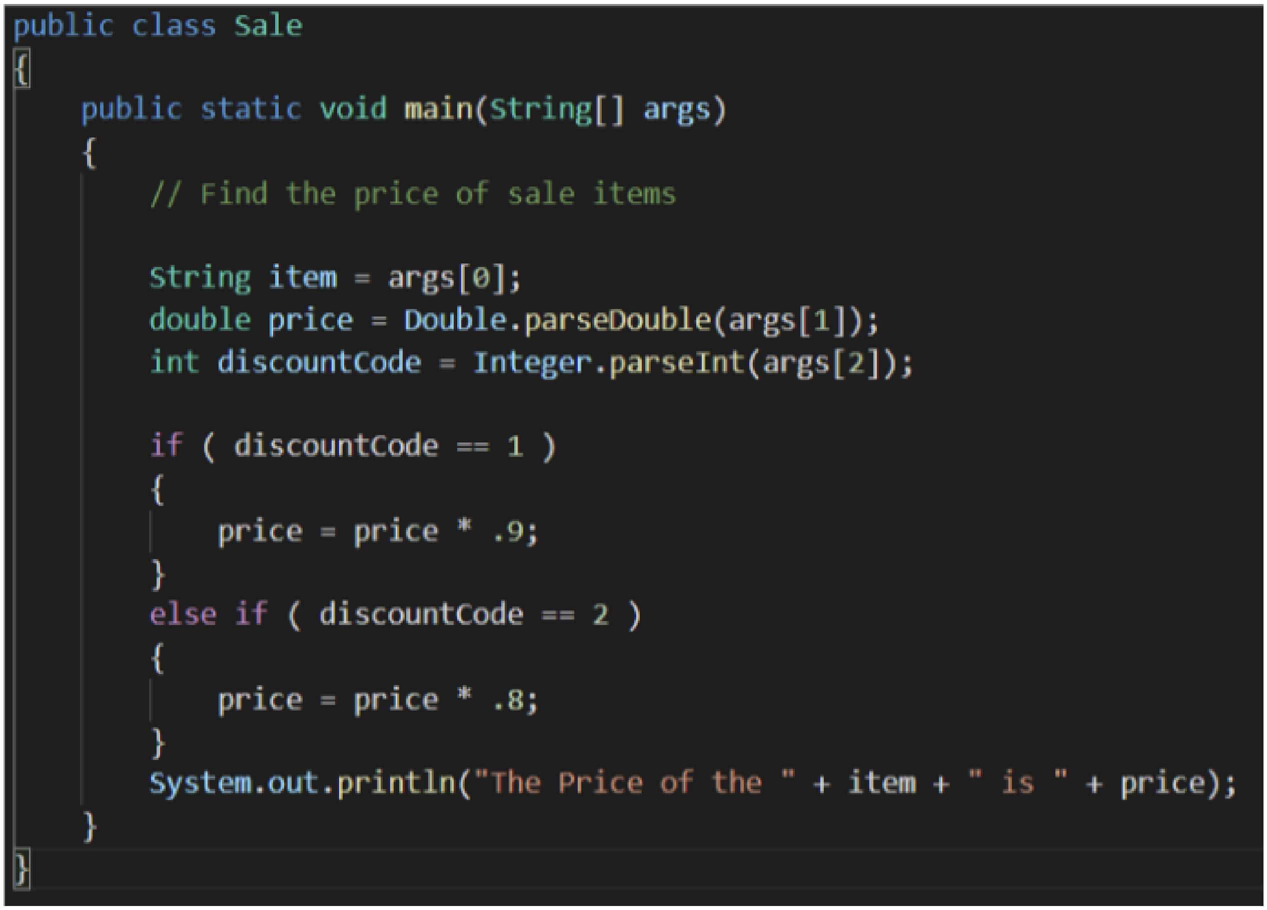
+ 0 pts incorrect

Q1 Problem 1: Algorithm Implementation and Basic Java

30 Points

Q1.1 The program Sale reads three arguments from the command-line: (1) a String with the description of an item, (2) a double with the price of the item, and (3) an integer discount code that will determine the amount of discount that will be applied to the item price. The discount code is 0, 1, or 2, where 0 means no discount, 1 means 10% discount, and 2 mean 20% discount. The program displays the item description and its final price after the discount is applied.

15 Points



```
public class Sale
{
    public static void main(String[] args)
    {
        // Find the price of sale items

        String item = args[0];
        double price = Double.parseDouble(args[1]);
        int discountCode = Integer.parseInt(args[2]);

        if ( discountCode == 1 )
        {
            price = price * .9;
        }
        else if ( discountCode == 2 )
        {
            price = price * .8;
        }
        System.out.println("The Price of the " + item + " is " + price);
    }
}
```

Which of the following statements execute the program properly? Choose correct or incorrect. **(5 points each)**

☑ java Sale teapot 12.20 1

☑ java Sale microwave 200 2

□ java Sale 88.19 blender 2

☑ java Sale potholders 15.00 0

□ java Sale crockpot 45.00

Q1.2 Translate the following pseudocode into a Java program called DivideItems. Assume that dividend and divisors are real numbers. Use the text box or a file upload for your answer.

15 Points

```
READ Dividend
READ Divisor
IF Divisor equals 0 THEN
   DISPLAY ERROR
ELSE
  COMPUTE Quotient AS Dividend / Divisor
  DISPLAY Quotient
ENDIF
```

```
public class DivideItems {

    public static void main (String [] args) {
        double Dividend = Double.parseDouble(args[0]);
        double Divisor = Double.parseDouble(args[1]);

        if (Divisor == 0) {
            System.out.println("ERROR");
        }
        else {
            double Quotient = Dividend/Divisor;
            System.out.println(Quotient);
        }
    }
}
```

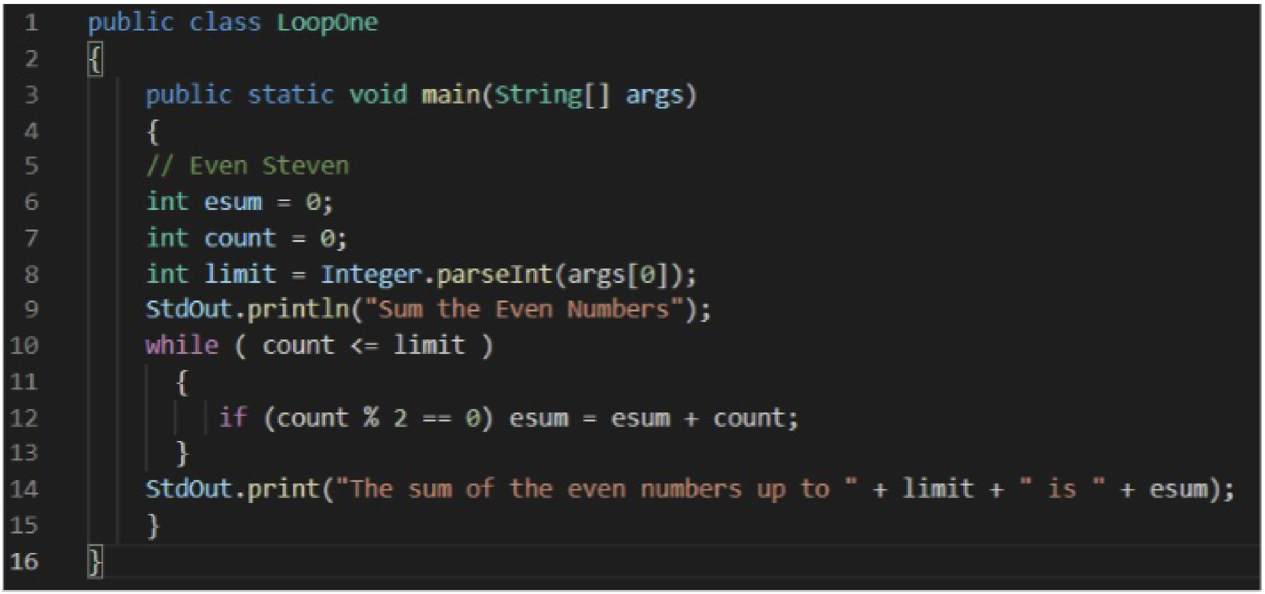
No files uploaded

Q2 Problem 2: Loops

30 Points

Q2.1 The following Java program is intended to take an integer limit from the command line and return the sum of all the even integers between 0 and the limit inclusive (we assume the input is an integer here). The code does not work as intended. Describe the changes to be made in the code so that the correct sum is calculated. Use the text box or a file upload for your answer.

15 Points



```
public class LoopOne
{
    public static void main(String[] args)
    {
    // Even Steven
    int esum = 0;
    int count = 0;
    int limit = Integer.parseInt(args[0]);
    StdOut.println("Sum the Even Numbers");
    while ( count <= limit )
      {
        if (count % 2 == 0) esum = esum + count;
      }
    StdOut.print("The sum of the even numbers up to " + limit + " is " + esum);
    }
}
```

This code results in an infinite loop. The variable "count" must be incremented at the end of the while loop in order for the correct sum to be calculated.

No files uploaded

Q2.2 Write a Java program that averages the multiples of 17 for the first two hundred thousand natural numbers. Use the text box or a file upload for your answer.
15 Points

```
public class Multiple {

    public static void main (String [] args) {
        double sum = 0;
        int count = 0;

        for (int i = 200000; i > 0; i--) {
            if (i%17 == 0) {
                sum += i;
                count++;
            }
        }
        double average = sum/count;
        System.out.println("The average is " + average);

    }
}
```

No files uploaded

Q3 Problem 3: Arrays

40 Points

Q3.1
10 Points

A. (2 points) Which of the following choices is the correct syntax for declaring and initializing an array **a** of five doubles?

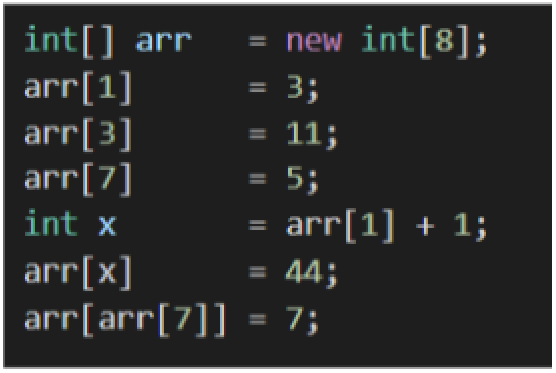
○ [] double a = [5] double;

○ double a[5] = **new** double[5];

○ double[5] a = **new** double[5];

● double[] a = **new** double[5];

B. (2 points) What does the array arr contain after the following code segment is executed? Show the content of the entire array for full credit. Use the text box or a file upload for your answer.



```
int[] arr    = new int[8];
arr[1]       = 3;
arr[3]       = 11;
arr[7]       = 5;
int x        = arr[1] + 1;
arr[x]       = 44;
arr[arr[7]] = 7;
```

{0, 3, 0, 11, 44, 7, 0, 5}

No files uploaded

C. (2 points) Assume that an array of **int nums** has been declared and initialized with integer values. Use the text box or a file upload for your answer.

Write a Java statement that assigns the value 10 to the first element in **nums**.

nums[0] = 10;

No files uploaded

D. (2 points) Write a segment of code that exchanges the first element of **nums** with the last element of **nums**. Your solution should work as intended for arrays of any size. Use the text box or a file upload for your answer.

```
int temp = nums[0];
nums[0] = nums[nums.length - 1];
nums[nums.length - 1] = temp;
```

No files uploaded

E. (2 points) Write a Java statement (one line of code) that declares and initializes the 1-dimensional integer array to the following values: 23, 42, 12, 6, 8. Use the text box or a file upload for your answer.

```
int[] arr = {23, 42, 12, 6, 8};
```

No files uploaded

Q3.2 The java program ArrayTest.java is intended to do the following:
15 Points

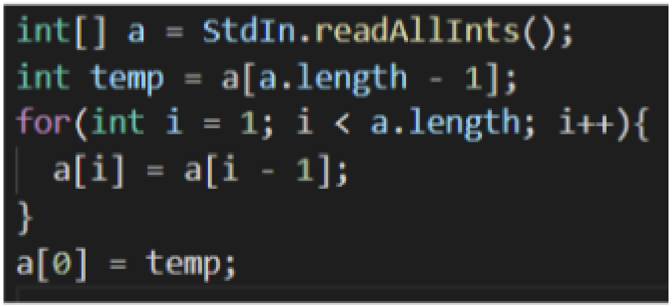
• Fill the array **a** with integer values input from the file ints.txt.
• (Shift-right) Shift the elements of **a** to the right one position and put the last element of array **a** in the first position in a (wrap around).
Example: if `a` **originally contained {1, 2, 3, 4} then the resulting array a would be {4, 1, 2, 3}**

A. (5 points) What command would be given on the command line to redirect the input to the program ArrayTest to come from the text file *ints.txt*? Use the text box or a file upload for your answer.

```
java ArrayTest < ints.text
```

No files uploaded

B. (5 points) The code segment below is intended to read ints.txt into the array **a**, and to complete the shift-right as explained above. The code does not work as intended.



```
int[] a = StdIn.readAllInts();
int temp = a[a.length - 1];
for(int i = 1; i < a.length; i++){
  a[i] = a[i - 1];
}
a[0] = temp;
```

If **a** was originally **{1,2,3,4,5,6}** What would the resulting array contain after the code segment above was executed? Use the text box or a file upload for your answer.

```
{6, 1, 1, 1, 1, 1}
```

No files uploaded

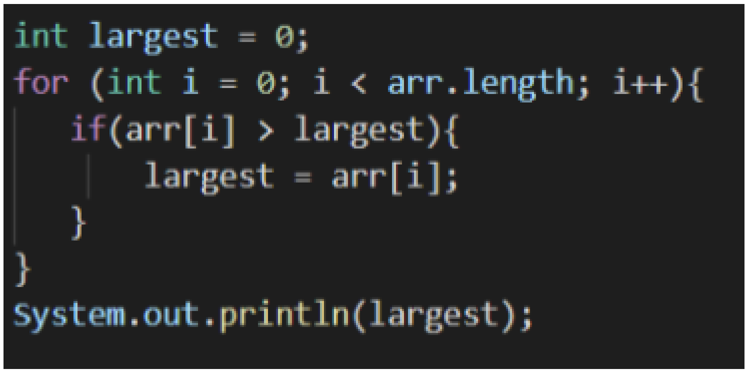
C. (5 points) Rewrite the code segment so that it performs the intended task (Shift the elements of a to the right one position and put the last element in the first position in the array (wrap around). Use the text box or a file upload for your answer.

```
int[] a = StdIn.readAllInts();
int[] b = new int[a.length};
b[0] = a[a.length - 1];
for (int i = 1; i < a.length; i++) {
b[i] = a[i - 1];
}
for (int 1 = 0; i < b.length; i++) {
a[i] = b[i];
}
```

No files uploaded

Q3.3
5 Points

The following segment of code is intended to find and display the largest value in an array of integers. It does not work as intended. Give **one example** of an array that would demonstrate that this code fails to display the largest value in the array. Use the text box or a file upload for your answer.



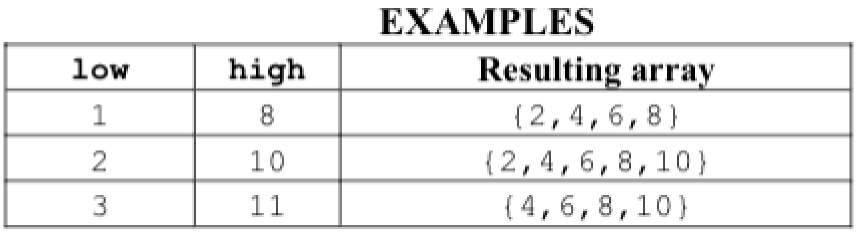
```
int largest = 0;
for (int i = 0; i < arr.length; i++){
    if(arr[i] > largest){
        largest = arr[i];
    }
}
System.out.println(largest);
```

```
int[] arr = {1, 1, 1}
```

No files uploaded

Q3.4
10 Points

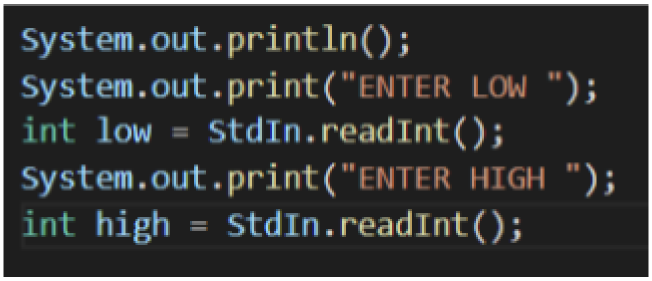
Write a Java code segment that will create an array evens and fill it with the even integers between low and high inclusive (where low and high are input by the user). The array you create must be the exact size that is needed to contain these even integers between low and high inclusive.



EXAMPLES

low	high	Resulting array
1	8	{2,4,6,8}
2	10	{2,4,6,8,10}
3	11	{4,6,8,10}

You can assume that low <= high
The code below will be executed to get input from the user.



```
System.out.println();
System.out.print("ENTER LOW ");
int low = StdIn.readInt();
System.out.print("ENTER HIGH ");
int high = StdIn.readInt();
```

Write your code to create and fill the array in the text box or upload a file with the code:

```
for (i = low; i <=high; i++) {
int a = low;
a++
if (a%2 == 0 ){
System.out.println(a);
}
}
```

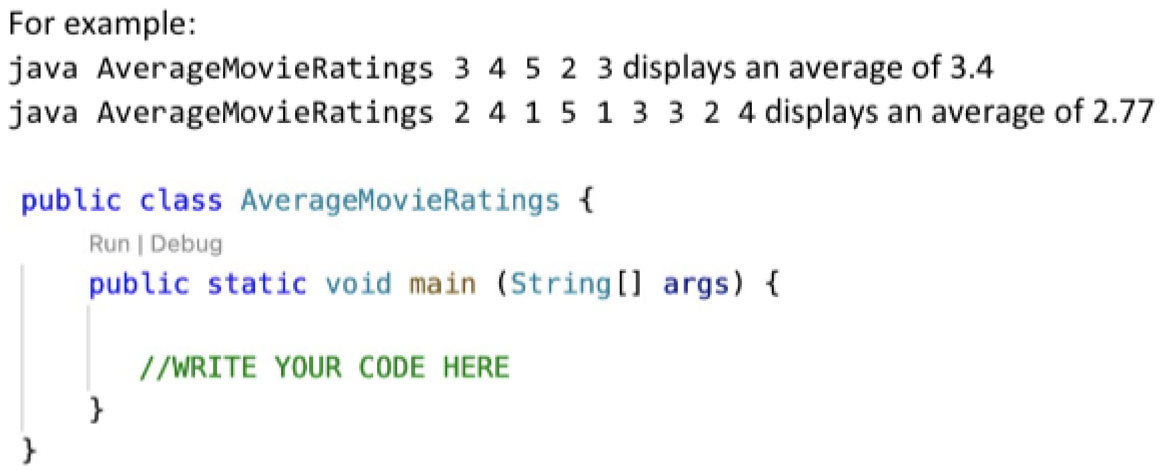
No files uploaded

Q4 Problem 4: Input and Output

30 Points

Q4.1
15 Points

A group of friends went to the movies to watch the latest Bond movie. After watching the movie each friend gave it a rating between 1 and 5 (5 means the movie was fantastic). Write the program AverageMovieRatings.java that receives as inputs through the command line the movie ratings from each friend, then prints the average of all friends' ratings. Note that the program is supposed to handle any number of friends. Use the text box or a file upload for your answer.



For example:
`java AverageMovieRatings 3 4 5 2 3` displays an average of 3.4
`java AverageMovieRatings 2 4 1 5 1 3 3 2 4` displays an average of 2.77

```java
public class AverageMovieRatings {
      Run | Debug
      public static void main (String[] args) {

          //WRITE YOUR CODE HERE
      }
}
```

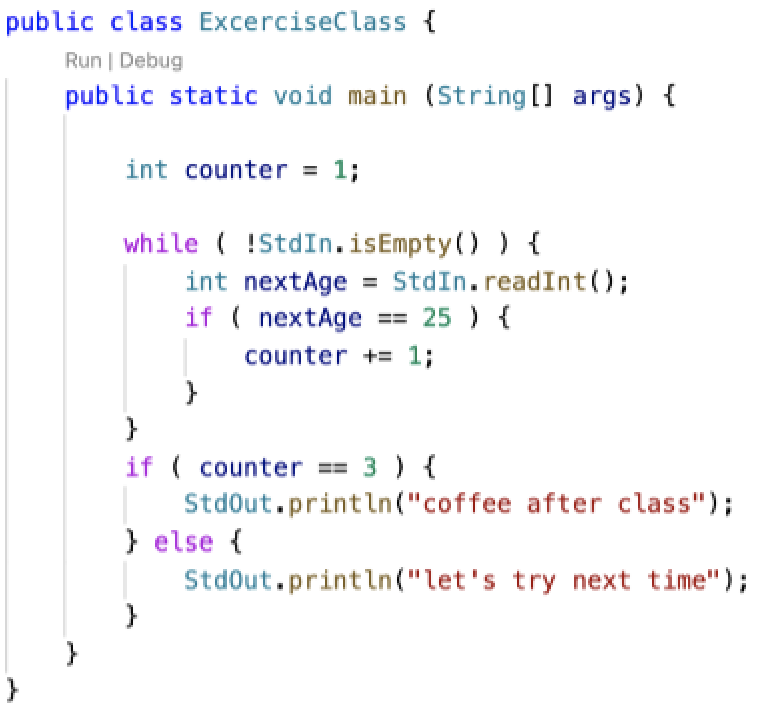
Use the text box or a file upload for your answer.

```java
public class AverageMovieRatings {
public static void main (String [] args) {
int arr[] = new int [args.length];
for (int i = 0; i < args.length; i++){
arr[i] = Integer.parseInt(args[i]);
}
int n = arr.length - 1;
double average = 0;
for (int i = 0; i < arr.length; i++){
average += arr[i];
}
System.out.println("Average: " + average/n);
}
}
```

No files uploaded

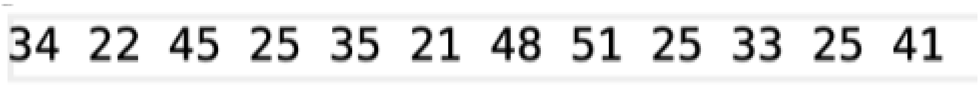
Q4.2
15 Points

A health club has a treat policy for all its exercise classes: if exactly 3 people in an exercise class are 25 years old then the whole group gets coffee after the class. The following program, which has a bug, is supposed to read the ages of all people in the class and then display “coffee after class” if exactly 3 people are 25 years old, otherwise it displays “let’s try next time”.

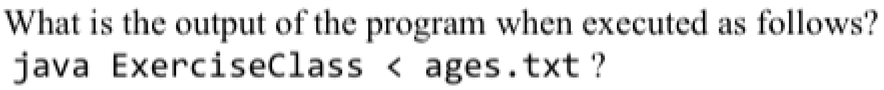


```
public class ExcerciseClass {
    Run | Debug
    public static void main (String[] args) {

        int counter = 1;

        while ( !StdIn.isEmpty() ) {
            int nextAge = StdIn.readInt();
            if ( nextAge == 25 ) {
                counter += 1;
            }
        }
        if ( counter == 3 ) {
            StdOut.println("coffee after class");
        } else {
            StdOut.println("let's try next time");
        }
    }
}
```

A. (5 points) Given the input file ages.txt (values in the file depicted below)
ages.txt



```
34 22 45 25 35 21 48 51 25 33 25 41
```



What is the output of the program when executed as follows?
`java ExerciseClass < ages.txt` ?

Use the text box or a file upload for your answer.

let's try next time

No files uploaded

B. (5 points) Without making any changes to the program, give an input file that would display “coffee after class”.

Use the text box or a file upload for your answer.

22 23 53 12 25 32 25 36

No files uploaded

C. (5 points) What is the bug in the program? How would you fix it?

Use the text box or a file upload for your answer.

The variable "counter" starts at value 1 which counts an extra person. It should start at 0.

No files uploaded