Hands-on Experiment # 6 : Worksheet (Iteration)

Section_____1_____ Date_____15/9/2020_____

Student ID _____6338110221_____ Name_____Nonthapat Kaewamporn_____

Part A: Loop Writing Practice

Convert the given `do-while` into `while` and `for` loop.

```
int k = 0;
int n = 800;
do {
    System.out.println(k);
    k++;
    n /= 2;
} while (n > 0);
```

List your `while` loop here.

```
int k = 0;
int n = 800;
while (n > 0)
{
        System.out.println(k);
        k++;
        n /= 2;
}
```

List your `for` loop here.

```
for (int n = 800, k = 0; n > 0; n /= 2, k++)
{
      System.out.println(k);
}
```

Program `ParityCheck.java` reads data from a text file given from command line and finds the parity of each number in the file. The output of the program is given below. Complete the methods: `countOnes()` and `parity()` using "loops" in `ParityCheck.java` to make the `ParityCheck` runs correctly.

An integer number has EVEN parity if sum of all binary representation bit is even number, and ODD parity otherwise. For example, a number 14 is '1110' in binary. Summation of all bits is 3. Therefore, 14 has ODD parity. A number 33 is '10001' in binary. Summation of all bits is 2. Therefore, 33 has EVEN parity.

Note: `Integer.toBinaryString()` convert an integer number to a binary String.

```
→  L06_sol java ParityCheck data.txt
Data read from: data.txt
123:
        Binary: 1111011
        # of 1's = 6
        parity: EVEN
2346:
        Binary: 100100101010
        # of 1's = 5
        parity: ODD
1285434:
        Binary: 10011100111010100111010
        # of 1's = 12
        parity: EVEN
25993669:
        Binary: 11000110010100000111000101
        # of 1's = 11
        parity: ODD
458299015:
        Binary: 11011010100010001011010000111
        # of 1's = 14
        parity: EVEN
→  L06_sol
```

Fill out the result with input from "`data.txt`"

Number read from file	Binary representation	How many 1's
123	1111011	6
2,346	100100101010	5
1,285,434	10011100111010100111010	12
25,993,669	11000110010100000111000101	11
458,299,015	11011010100010001011010000111	14

List your method `countOnes()` here

```java
public static int countOnes(String binaryString) {
    for(int i = 0; i < binaryString.length(); i++) {
        if(binaryString.charAt(i) == '1') n++;
    }
    return n;
}
```

List your method `parity()` here

```java
public static String parity(String binaryString) {
    if(countOnes(binaryString)%2 == 0) return "EVEN";
    return "ODD";
}
```

Run your program and capture the output screen here. The output should similar the following example but show that actual data read from '`data.txt`'

```
C:\Users\User\Desktop\Work\Class Materials\Com Prog\HW6>java ParityCheck data.txt
Data read from: data.txt
123:
        Binary: 1111011
        # of 1's = 6
        parity: EVEN
2346:
        Binary: 100100101010
        # of 1's = 5
        parity: ODD
1285434:
        Binary: 100111001110100111010
        # of 1's = 12
        parity: EVEN
25993669:
        Binary: 110001100101000011000101
        # of 1's = 11
        parity: ODD
458299015:
        Binary: 11011010100010001011010000111
        # of 1's = 14
        parity: EVEN
```

Part B: Square Root Estimation

We can use bisection method to estimate the value of $\sqrt{a}$ as following:

1. Let *lower = 0, upper = a*
2. Begin with the estimated value, *x*, is the middle of [*lower, upper*]
3. Repeat the following steps until x^2 is closed enough to *a* ("closed enough" when $|a - x^2| \leq 10^{-10} max(a, x^2)$)
    a. if x^2>a then change the range to [lower, x]
    b. if x^2<a then change the range to [x, upper]
    c. x is the middle value
4. x is the estimated value of $\sqrt{a}$

Complete the method `bisection()` in `Bisection.java` to estimate the square root of values reading from "`bisection.in`". The output of the program should be similar to the picture below.

```
➜  L06_sol java Bisection
Square root of 2.00000 is 1.4142135623842478, expected: 1.4142135623842478
Square root of 3.00000 is 1.7320508075645193, expected: 1.7320508075645193
Square root of 4.00000 is 2.0000000000000000, expected: 2.0000000000000000
Square root of 9.00000 is 3.0000000000873115, expected: 3.0000000000873115
Square root of 20.0000 is 4.4721359550021590, expected: 4.4721359550021590
Square root of 100.000 is 9.9999999998544800, expected: 9.9999999998544800
Square root of 144.000 is 12.000000003492460, expected: 12.000000003492460
➜  L06_sol
```

Run your program and capture the output screen

```
C:\Users\User\Desktop\Work\Class Materials\Com Prog\HW6>java Bisection
Square root of 2.00000 is 1.4142135623842478, expected: 1.4142135623842478
Square root of 3.00000 is 1.7320508075645193, expected: 1.7320508075645193
Square root of 4.00000 is 2.0000000000000000, expected: 2.0000000000000000
Square root of 9.00000 is 3.0000000000873115, expected: 3.0000000000873115
Square root of 20.0000 is 4.4721359550021590, expected: 4.4721359550021590
Square root of 100.000 is 9.9999999998544800, expected: 9.9999999998544800
Square root of 144.000 is 12.000000003492460, expected: 12.000000003492460
```

List your method `bisection()` here

```java
public static double bisection(double a) {
    double x = 0, l = 0, u = a;
    while(Math.abs(a - x*x) >= Math.pow(10,-10) * Math.max(a,x*x)){
        x = (l+u)/2;
        if(x*x > a) u = x;
        else l = x;
    }
    return x;  // do not modify this
}
```

Part C: Text File Processing

"`sheet1.in`" is a text file contains multiple choice answers of students. The first line is the correct answers. The remaining lines are students name, follow by at least one space and their answers as show below.

```
ABCDEABCDEEDCBAEDCBA
Bank ABCDDDDCDEEDCBAEDCBA
Lin ABBDEBBCDEBDCBBEDBBA
Pat AACCEEBBDDEECCAADDBB
Grace BBDDAACCEEDDBBEECCAA
```

Complete the program `MultipleChoiseApp.java` to read input data file as command line argument. There are two data files given, `sheet1.in` and `sheet2.in`. The expected output is also given for you in `sheet1.out` and `sheet2.out` to check that your program executes correctly. Your program should produce the output similar to the given picture below. The first line shows total number of questions follow by all correct answer. Each of the remaining lines shows student's name, his/her score and their marked answers. (The question that is correct, show as original. The question that is wrong, show as '-'.)

```
→  L06_sol java MultipleChoiceApp sheet1.in
         [20]    {ABCDEABCDEEDCBAEDCBA}
Bank     [17]    {ABCD---CDEEDCBAEDCBA}
Lin      [15]    {AB-DE-BCDE-DCB-ED-BA}
Pat      [10]    {A-C-E-B-D-E-C-A-D-B-}
Grace    [10]    {-B-D-A-C-E-D-B-E-C-A}
→  L06_sol
```

Run your program with "`sheet1.in`" and capture the output screen here.

```
C:\Users\User\Desktop\Work\Class Materials\Com Prog\HW6>java MultipleChoiceApp sheet1.in
                [20]            {ABCDEABCDEEDCBAEDCBA}
Bank            [17]            {ABCD---CDEEDCBAEDCBA}
Lin             [15]            {AB-DE-BCDE-DCB-ED-BA}
Pat             [10]            {A-C-E-B-D-E-C-A-D-B-}
Grace           [10]            {-B-D-A-C-E-D-B-E-C-A}
```

Run your program with "`sheet2.in`" and capture the output screen here.

```
C:\Users\User\Desktop\Work\Class Materials\Com Prog\HW6>java MultipleChoiceApp sheet2.in
                [50]            {ABCDECDEABEDBACBAACEBCDEACBBDEEDCBAABCDEECECBDBDDA}
Bank            [10]            {A-------A---B--B-----C---C-----D------D-E-E-------}
Lin             [9]             {A-------A----A--AA------A---------AA-------------A}
Pat             [17]            {ABCDE--------------E------B-DE--C--ABCDE------B-D-}
Grace           [9]             {ABCDE-D-------------------------C----C-----------A}
Joey            [15]            {ABCDE---AB-D--CBA-C-----A--------B-----------D----}
Tony            [13]            {--C---D---ED------------A-----EDCBA--C--E--------A}
John            [11]            {A----CD---E---C----------C---EE-C---B----------D--}
```

*** Submission instructions ***

Combine the following files into ONE .zip file, “assignment-6.zip”.

1. This worksheet file saved as “Lab06-2020.pdf”.
2. ParityCheck.java
3. Bisection.java
4. MultipleChoiceApp

Submit the zip file to http://www.myCourseVille.com (Assignments > Assignment 6) before midnight of three days after your lecture.