

EECS 1510: Object Oriented Programming
Project 6 – Methods, Arrays, and Strings
190 Points Due in class Tuesday March 27, 2018

Submission Procedure. For each program, print *two* sample runs, one using the values from a sample dialog below, the other using values from some “special” case.

Grading: You are not required to use the program standards for this project. However, projects of high quality will be awarded 10 to 20 points extra credit

Program 1 (30 points) Valid Phone Numbers

Phone number can have one of several valid formats: In particular, strings like

419-460-1212 (419) 460-1212 460-1212

are valid, but strings like

419-460 (419) 460-a321 46012-12

are not. Write a program to read in a string and check whether it has the format

ddd-ddd-dddd

where each d is a digit. (Only this format need to be checked.)

Program 2 (40 points) BinaryConversion.java

Consider the following dialog:

```
Enter a binary number: 1110
Conversion to decimal: 14
Enter a binary number: 10010000
Conversion to decimal: 144
Enter a binary number: -1
All set !
```

Write the application implied above. In particular, the program will read a sequence of binary strings and convert each one to a decimal integer. The program will terminate when the string -1 is given. You must do the conversion by hand, and NOT use the predefined functions in Java for the wrapper class Integer.

You must use a function for the conversion to decimal, where the parameter to the function is the binary string, and the return value is the equivalent decimal integer.

```
public static int binaryToDecimal (String binaryString)
```

Program 3 (30 points) Reverse.java

Write a program that reads in 8 positive integers, e.g.

73 95 61 21 90 85 14 78

into an array. The program then prints the values in reverse order as well as the average (to one decimal place). For example,

```
Please enter 8 positive integers: 73 95 61 21 90 85 14 78
```

```
The values in reverse order are
    78  14  85  90  21  61  95  73
The average is 517/8 = 64.6
```

You must use a function a function for computing the reverse, passing the array as an argument. Thus the function header will be similar to

```
void reverse(int[] A)
```

Program 4. ScanningText.java (40 points) Write a program to read in a line of text and give as output the number of letters, the number of string tokens, and a table giving the frequency of each letter. For example,

Enter a single line of text:

```
Now is the time for all good men to come to the aid of their country.
```

```
The line contains 53 letters.
The line contains 16 string tokens.
The frequency of letters is
```

```
A -- 2
C -- 2
D -- 2
E -- 6
F -- 2
G -- 1
H -- 3
I -- 4
L -- 2
M -- 3
N -- 3
O -- 9
R -- 3
S -- 1
T -- 7
U -- 1
W -- 1
Y -- 1
```

The table of letter frequencies must use capital letters (as above), but can be in more than one column. Use an array to store the letter frequencies. Name the program **ScanningText.java**.

Hint: Read the line of text into a string variable using the `nextLine` method,

```
String line;
line = input.nextLine();
```

and later use

```
int ch, count;
for (ch='a'; ch<='z'; ch++){
    -- Count the number of occurrences of ch in the line
    -- Print the count if > 0
}
```

Program 5 (50 points) FastestRunner.java A group of 16 students decided to run in the Columbus Marathon. Their names and times (in minutes) are below:

Name	Time (minutes)
Elena	341
Thomas	273
Hamilton	278
Suzie	329
Phil	445
Matt	402
Alex	388
Emma	275
John	243
James	334
Jane	412
Emily	393
Daniel	299
Neda	343
Aaron	317
Kate	265

Find the fastest runner.

In particular, write a program as follows. Print the list of runners and times as above. Then print the name of the fastest runner and his/her time (in hours and minutes).

The names and running times are to be entered into the program as shown below. The program should have a method that takes as input an array of integers and returns the index corresponding to the person with the lowest time. The program should apply this method to the array of running times to find the fastest runner.

Extra Credit (10 points): Print the list of runners and times nicely formatted in columns as displayed above. One option is to use the `printf()` method.

Here is some program code to get started:

```

class Marathon {
public static void main (String[] arguments){
    final int numRunners = 16;
    String[] names ={"Elena", "Thomas", "Hamilton", "Suzie", "Phil", "Matt",
        "Alex", "Emma", "John", "James", "Jane", "Emily", "Daniel",
        "Neda", "Aaron", "Kate"};

    int[] times ={341, 273, 278, 329, 445, 402, 388, 275, 243, 334, 412,
        393, 299,343, 317, 265};
    . . .

    for (int i = 0; i < numRunners; i++) {
        System.out.println(names[i]+ ": " + times[i]);
    }
    . . .
}

```