King Fahd University of Petroleum and Minerals

###### Information and Computer Science Department

**ICS 102: Introduction to Computer Programming**

**Summer Semester 2018-2019 (Term 183)  
Lab Test #1**

Consider a forecasting station that has an archive of files containing readings of temperature for different number of days. The data is not uniform, as some days may include more readings than others, while it is possible to have a day with no reading. A sample text file “weather.txt” is shown below.

Wed 12/11/2001 35 36 37 37 36 34 33 32 30 29

Thu 13/11/2001 25 26 27 27 26 24 20 19

Fri 14/11/2001 45 46 47 47 43 42 40 39

Sat 15/11/2001

Sun 16/11/2001 35 36 37 37 36 34 33

Mon 17/11/2001 37 36 34 33 32 30 29

Tue 18/11/2001 35 36 37 37 36 34 33 32 30

You may assume that each line contains at least the day and the date. However, it may include zero or more readings per day.

Develop a java program that reads “weather.txt” and produces a summary report text file “report.txt” that does the following: For each line, it prints the day *in capital letters*, followed by the date, followed by the number of readings per day, followed by the minimum reading and finally the maximum reading. If a specific day does not contain any reading, it should display “No Readings Available”. The report contains a header line in the beginning indicating this information. A sample “report.txt” for the above “weather.txt” file is shown below.

Day Date #Readings Min Max

WED 12/11/2001 10 29 37

THU 13/11/2001 8 19 27

FRI 14/11/2001 8 39 47

SAT 15/11/2001 No Readings Available

SUN 16/11/2001 7 33 37

MON 17/11/2001 7 29 37

TUE 18/11/2001 9 30 37

This report contains 7 readings

King Fahd University of Petroleum and Minerals

###### Information and Computer Science Department

**ICS 102: Introduction to Computer Programming**

**Summer Semester 2018-2019 (Term 183)  
Midterm (Part I)**

A *Perfect number* is any number whose sum of its positive divisors, excluding the number itself, is equal to that number. For example, 28 is a perfect number because 28 is divisible by 1, 2, 4, 7, 14 and 28 and the sum of these values are: 1 + 2 + 4 + 7 + 14 = 28 (Remember, we have to exclude the number itself 28. That’s why we haven’t added 28 here). Some of the perfect numbers are 6, 28, 496, 8128, 33550336 and so on.

An *Armstrong Number* is an integer such that the sum of the cubes of its digits equals the number itself. For example, 153 is an Armstrong number because .

Write a program that prompts the user to input a maximum number , where , followed by the following menu of options:

1- Perfect Numbers

2- Armstrong Numbers

3- Exit

If the user enters 1, your program should display on the screen all numbers that are perfect in the range . If the user enters 2, your program should display on the screen all numbers that are Armstrong numbers in the range . If the user enters 3, the program terminates.

Notes:

1. If the user enters a non-integer value, the program should display an appropriate message and terminate gracefully.
2. If the user enters a value other than 1, 2 or 3, the program should display an appropriate message and seek the user’s input again.

Sample Run 1:

Enter an integer n such that 100 <= n <= 100000

1rt

Non integer value entered!....exiting....

Sample Run 2:

Enter an integer n such that 100 <= n <= 100000

5

The value 5 is not in the specified range

Enter an integer n such that 100 <= n <= 100000

200

1- Perfect Numbers

2- Armstrong Numbers

3- Exit

Enter an option: a

Non integer value entered!....exiting....

Sample Run 3:

Enter an integer n such that 100 <= n <= 100000

1000

1- Perfect Numbers

2- Armstrong Numbers

3- Exit

Enter an option: 1

list of Perfect numbers between 2 and 1000

6

28

496

1- Perfect Numbers

2- Armstrong Numbers

3- Exit

Enter an option: 5

Invalid Option. Please try again

1- Perfect Numbers

2- Armstrong Numbers

3- Exit

Enter an option: 2

list of Armstrong numbers between 100 and 1000

153

370

371

407

1- Perfect Numbers

2- Armstrong Numbers

3- Exit

Enter an option: 3

Exiting