

## QUIZ 1

**TAs :** Nebi YILMAZ

**Due Date :** 19.03.2023 (23:59:59)

### 1 Introduction

In this experiment, you are going to practice on control and loop statements, arrays and functions of Java.

### 2 Experiment

You are expected to develop a program that performs some mathematical operations. These operations are;

- Calculating Armstrong numbers
- Calculating Emirp numbers
- Generating number sequences in either ascending or descending order.

#### 2.1 Calculating Armstrong numbers

An n-digit number is an Armstrong number if the sum of the nth power of the digits is equal to the original number. Here are some example Armstrong numbers;

<b>1634</b> (here n = 4)	<b>123</b> (here n = 3)
$= 1^4 + 6^4 + 3^4 + 4^4$	$= 1^3 + 2^3 + 3^3$
$= 1 + 1296 + 81 + 256$	$= 1 + 8 + 27$
$= 1634$	$= 36$

By the selection of this operation from input file, your program is going to get an input value from the file and list all Armstrong numbers up to that value (including that number).

#### 2.2 Calculating Emirp numbers

A number is called an emirp number if we get another prime number on reversing the number itself. In other words, an emirp number is a number that is prime forwards or backward. It is also known as twisted prime numbers.

Suppose, we want to check the number 13 is emirp or not.

We know that 13 is a prime number means that divisible by 1 and self only. On reversing the number, we get 31 which is another prime number. Therefore, 13 and 31 both are Emirp numbers.

By the selection of this operation from input file, your program is going to get an input value from the file and list all Emirp numbers up to that value (including that number).

## 2.3 Calculating Abundant numbers

In number theory, an abundant number is a number for which the sum of its proper divisors is greater than the number.

Suppose, we want to check the number 48 is abundant or not.

The proper divisors of 48 (without itself (i.e., 48)) are  $1+2+3+4+6+8+12+16+24=76$

Therefore, 48 is called an abundant number because it is less than the sum of its divisors (without itself (i.e., 48)). (76 is greater than 48)

By the selection of this operation from input file, your program is going to get an input value from the file and list all Abundant numbers up to that value (including that number).

## 2.4 Sorting

If the sorting command comes, an array of integers is going to be generated from numbers that will be given by the file. The array is going to be in either ascending or descending order according to command in the file. The commands for sorting will be like "Ascending order sorting" or "Descending order sorting". While the each number comes from the file, new element is going to be inserted to true position of the array according to command type and the number sequence will be printed. -1 means the array generation is finished.

## 2.5 Exit

If the command is "Exit", the program will terminate. The program will continue to run until the "Exit" command.

## 2.6 Example input/output

Example input/output will be given in piazza. Please follow the Piazza.

## Grading Policy

Task	Point
Submit	1
Correct output.txt	99
Total	100

## Submit Format

File hierarchy must be zipped before submitted (Not .rar, only .zip files are supported by the system)

- <studentid>.zip
  - <src>
    - Main.java, \*.java

## Execution and Test

The input file is going to be given as program arguments. In order to test your program, you should follow the following steps:

- Upload your java files to your server account (dev.cs.hacettepe.edu.tr)
- Compile your code (javac \*.java, or javac Main.java)
- Run your program (java Main input.txt)
- Control your output file (output.txt).

## Notes

- Do not miss the submission deadline.
- Save all your work until the assignment is graded.
- Compile your code on DEV server before submitting your work to make sure it compiles without any problems on our server.
- You can ask your questions through course's piazza group and you are supposed to be aware of everything discussed in the piazza group. General discussion of the problem is allowed, but DO NOT SHARE answers, algorithms, source codes and reports.
- All assignments must be original, individual work. Duplicate or very similar assignments are both going to be considered as cheating.