



Exercise 1

1. Some more recent board-games use dice with different number of sides, 4,6,8,10,12,20, ... Here, each side contains its corresponding number. For instance, if we have a 4-sided dice, then side 1 will have 1 dot, the second will have 2 dots, the third three, and the fourth four dots. Write a program **Dice.java** that takes the number of sides as input from the command line, and then simulates a random roll with that die and prints the result. Your program should work for all whole positive number, not just for the 6 values listed above.

Exercise 2

2. Write a program **Even.java** that reads an integer number from the command line and prints “true” if the number is even. Make sure your program still works if the number entered was a negative integer.

Exercise 3

3. Write a program **ListDigits.java** that reads in an integer from command line and breaks it into a sequence of individual digits in reverse order. Example:

```
C:\> java ListDigits 16384
4
8
3
6
1
```

Exercise 4

4. Write a program **DicePair.java** that simulates rolling a pair of dice until the total on the dice comes up to be a given positive integer entered from the command line. Your program should include a static method, which takes the number entered as an input and returns the total number of rolls. The program should then print the total number of rolls returned.

Every time you roll the dice repeatedly, trying to get a given total, the number of rolls it takes can be different. The question of the average number of rolls naturally arises. Write a static method that performs the experiment of rolling to get a given total 10,000 times. The desired total is a parameter to this method. The average number of rolls is the return value. Each individual experiment should be done by calling the method you wrote in the first part. Now, write a main program that will call your methods once for each of the possible totals (2,3, ...,12). It should make a table of the results:

Total on Dice	Average Number of Rolls
-----	-----
2	35.8382
.	.

Exercise 5

5. Write a method `mangle` that accepts a string `str`, a character `c` and an integer `n` as arguments, such that the string `str` contains at least `n` times the character `c`. The method returns the resulting string after removing the first `n` occurrences of the character `c` from the original string `str` (the method should not print anything to the console). The signature of the method should be:

```
public static String mangle(String str, char c, int n)
```

Use this method to write a program `StringMangle.java` that takes as command line arguments an integer `N` followed by `N` tuples; each tuple consists of a string, a character and an integer (`str`, `c`, `n`). The program should print each of the strings after removing the first `n` occurrences of the character `c` from the corresponding string `str`.

Sample Run:

```
> java StringMangle 3 test t 1 ahaaaamad a 4 testtt t 2  
est hamad estt
```

Exercise 6

6. Write a program `NumStrings.java` that receives two strings through the command line as input and then prints out the number of times the second string occurs as a substring in the first.

Submission Instructions

- Your submission must consist of a single zip folder that contains .java files only (`Dice.java`, `Even.java`, `ListDigits.java`, `DicePair.java`, `StringMangle.java`, `NumStrings.java`). No additional files should exist in the .zip folder.
- Give meaningful names to your methods and variables in your code.
- Include a comment at the beginning of your program with basic information about yourself and a description of the program. Include also a comment at the start of each method.
- The name of the zip file must adhere to the following naming convention `s#_A$_netid`, where `#` stands for your section number (between 1 and 12), `$` stands for the assignment number, and `netid` stands for your AUBnet user name. For example, if your AUBnetid is `abc65` and you are in section 4, you should submit Assignment 6 using following file: `s4_A6_abc65.zip`. The zip files will be processed automatically so please make sure you use this naming convention.
- **Failing to follow these guidelines will result in deducting marks form your grade.**