Exercise 1

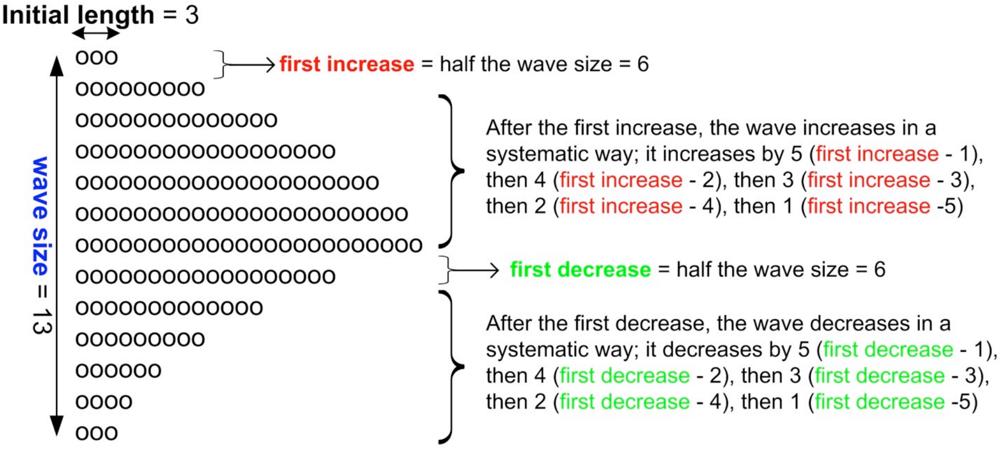
Write a program, **`Waves.java`**, that asks the user for the following input using the Scanner object: 1 string, which represents the symbol of the wave, and 3 integers: 1) the *numbers of waves*; 2) the *initial length of the wave*; 3) the *size of the wave*. The wave behavior is summarized as follows:

a) The wave starts with an initial length
b) The length of the wave suddenly increases by a value equivalent to the size of the wave divided by 2, i.e., increases by half the wave size
c) After this sudden increase, the wave continues to grow in a systematic way as shown in the figure below.
d) After the peak increase, the wave starts to decrease. The first decrease is equivalent to half the wave size.
e) After the first decrease, the wave continues to decrease in a systematic way until it reaches its initial length.
f) At this point, the second wave will start in exactly the same way.

The figure below shows the behavior of one wave having an initial length of 3 and a size of 13:

- Symbol = “o”
- Number of waves = 1
- Initial length = 3
- Wave size = 13 (the size is usually an odd number)

You should write the appropriate static method(s) (with the appropriate parameters) to show structure and remove redundancy in your solution.



```
ooo
ooooooooo
oooooooooooooo
oooooooooooooooooo
ooooooooooooooooooooo
ooooooooooooooooooooooo
oooooooooooooooooooooooo
oooooooooooooooooo
ooooooooooooo
ooooooooo
oooooo
oooo
ooo
```

first increase = half the wave size = 6

After the first increase, the wave increases in a systematic way; it increases by 5 (first increase - 1), then 4 (first increase - 2), then 3 (first increase - 3), then 2 (first increase - 4), then 1 (first increase -5)

first decrease = half the wave size = 6

After the first decrease, the wave decreases in a systematic way; it decreases by 5 (first decrease - 1), then 4 (first decrease - 2), then 3 (first decrease - 3), then 2 (first decrease - 4), then 1 (first decrease -5)

There are three sample runs provided on three separate pages

Sample Run 1 (number of waves=3, initial wave length=3, wave size=13):

```
> java Waves
Please enter the symbol: o
Please enter the no. of waves, initial wave length, and wave size: 3 3 13
○○○
○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○
○○○○○○○○○
○○○○○○
○○○○
○○○
○○○
○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○
○○○○○○○○○
○○○○○○
○○○○
○○○
○○○
○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○
○○○○○○○○○
○○○○○○
○○○○
○○○
```

Sample Run 2 (number of waves=4, initial wave length=5, wave size=9):

```
> java Waves
Please enter the symbol: o
Please enter the no. of waves, initial wave length, and wave size: 4 5 9
○○○○○
○○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○
○○○○○○○○
○○○○○○
○○○○○
○○○○○
○○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○
○○○○○○○○
○○○○○○
○○○○○
○○○○○
○○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○
○○○○○○○○
○○○○○○
○○○○○
○○○○○
○○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○
○○○○○○○○
○○○○○○
○○○○○
```

Sample Run 3 (number of waves=3, initial wave length=3, wave size=11):

```
> java Waves
Please enter the symbol: o
Please enter the no. of waves, initial wave length, and wave size: 3 3 11
○○○
○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○
○○○○○○○○○
○○○○○○
○○○○
○○○
○○○
○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○
○○○○○○○○○
○○○○○○
○○○○
○○○
○○○
○○○○○○○○
○○○○○○○○○○○○
○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○○○○○○
○○○○○○○○○○○○○
○○○○○○○○○
○○○○○○
○○○○
○○○
```

Exercise 2

Write a program, **Encrypt.java,** that asks the user to enter using the Scanner object one string that could contain spaces and encrypt that string by first shifting each character one position to the right in the ASCII table (**A** is replaced with **B**, **a** is replaced with **b**, **B** is replaced with **C**, etc.) then reversing the shifted string. Your program must contain three methods:

```
public static void main(String[] args)
public static String reverse(String text)  //returns the string reversed
public static String encrypt(String text)  /*returns the string encrypted (each character is
                                             shifted one position in the ASCII table then
                                             reversed)*/
```

Coding Specifications (you must abide by them):

1. If the user supplies more than one command line argument you need to treat them all as one string
2. `main()` calls `encrypt()`, it does not call `reverse()`
3. `encrypt()` calls `reverse()` before returning the string
4. The printing of the string happens in `main()`

Sample Run 1 (input is one string):

```
> java Encrypt
Please enter a sentence or word: ABCDEF
GFEDCB
```

Explanation:

1. First step is to shift each character by one position to the left, so:

   ABCDEF

   becomes:

   BCDEFG

2. Then reversing the shifted string gives:

   GFEDCB

Sample Run 2 (input is multiple strings that are to be treated as one):

```
> java Encrypt
Please enter a sentence or word: Hello World
pmmfI emspX
```

Explanation:

1. First step is to shift each character by one position to the left, so:

   Hello World

   becomes:

   Ifmmp Xpsme

2. Then reversing the shifted string gives:

   pmmfI emspX

Exercise 3

Write a program, **`Random.java`,** that would ask the user to enter using the Scanner object two integers (min and max), then output N number of lines (N is a random integer between min and max inclusive), where each line is a string of random lowercase characters of length M (M is a random integer between min and max inclusive).

Notes:

1. You must print the line numbers as shown in the sample run (i.e. the line number is part of your output)
2. Your random strings will differ than those printed here
3. Your number of lines is very likely to differ for the same min and max provided in the sample run

Sample Run (min=5, max=10):

```
> java Random
Please enter min and max: 5 10
1: qqoziiqd
2: hvhjp
3: saxqsiaeeu
4: naqwb
5: bgwhzx
6: eabmggkxq
7: pmjvfl
```

Explanation:

- The output here was 7 lines (a number between 5 and 10 inclusive)
- Each line consists of random lower case characters
- Each line’s length is between 5 and 10 characters inclusive
  line 1 has a length of 8
  line 2 has a length of 5
  line 3 has a length of 10
  line 4 has a length of 5
  line 5 has a length of 6
  line 6 has a length of 9
  line 7 has a length of 6

Exercise 4

Write a program **Cards.java** that takes as input from the user a playing card in the shorthand notation described below, and then prints the full description of the card.

Denomination	***Value***
A	Ace
2, …, 10	2, …, 10
J	Jack
Q	Queen
K	King

Suit	***Value***
D	Diamonds
H	Hearts
S	Spades
C	Clubs

Examples:

```
> java Cards
Enter the card value: AS
You selected: Ace of Spades

> java Cards
Enter the card value: 2H
You selected: 2 of Hearts

> java Cards
Enter the card value: QK
You selected: Queen of Unknown Suit

> java Cards
Enter the card value: VH
You selected: Unknown Denomination of Hearts
```

Exercise 5

Julio Cesar Chavez Mark VII is an interplanetary space boxer, who currently holds the championship belts for various weight categories on many different planets within our solar system. However, it is often difficult for him to recall what his "target weight" needs to be on earth in order to make the weight class on other planets. Write a program **Chavez.java** to help him keep track of this.

The program should ask the user (Julio) using Scanner what his earth weight is, and to enter a number for the planet he wants to fight on. It should then compute his weight on the destination planet based on the table below:

#	**Planet**	**Relative gravity**
1	Venus	0.78
2	Mars	0.39
3	Jupiter	2.65
4	Saturn	1.17
5	Uranus	1.05
6	Neptune	1.23

So, for example, if Julio weighs 128 lbs. on earth, then he would weigh just less than 50 lbs. on Mars, since Mars' gravity is 0.39 times earth's gravity: (128 * 0.39 is 49.92).

The following is a sample output:

```
> java Chavez
Please enter your earth weight in pounds: 128
I have information for the following planets:
   1. Venus    2. Mars     3. Jupiter
   4. Saturn   5. Uranus   6. Neptune
Which planet are you visiting? 2
Your weight on Mars would be: 49.92 pounds.
```

Exercise 6

Write a program **`Salary.java`** that asks the user to enter the annual salary of an employee for *N* years, such as *N* is specified as a command line argument. Your program should then:

- Calculate and display the average income of the *N* years;
- check If the average value:
  - ○ is less than or equal to $10000, then you should display a message saying: "`Your income is low`"
  - ○ is between $10000 and $15000, then you should display a message saying: "`Your income is OK`"
  - ○ is greater than $15000, then you should display a message saying: "`Your income is high`"

`Sample Run`

```
> java Salary 4
```

```
Enter your salary for the last 4 years: 11000 7000 11500 11000
Your average income is $10125
Your income is OK
```

Submission Instructions

- Your submission must consist of a single zip folder that contains .java files only (**`Waves.java, Encrypt.java, Random.java, Cards.java, Chavez.java,`** and **`Salary.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 7 using following file: s4_A7_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.**