Grade 12 Review and Intro to Java

Level 0 – Transistor Gain

New Concepts

- output(System.out.println), input(Scanner), Variables and Math
- Advanced: loops

Program Requirements

A transistor's current gain is calculated using the formula β = IC / IB, where β (Beta) is the current gain, IC is the collector current, and IB is the base current.

Write a program that prompts for Base and Collector currents, and outputs the current gain.

Example Run

Enter the amount of collector current (in Amperes): 0.1

Enter the amount of base current (in Amperes): 0.001

The transistor current gain is 100.

Bonus Exercise (figure out loops)

Modify the program so it continuously runs until 0 (zero) is entered for the collector current.

Level 1 - Rock, Paper, Scissors

New Concepts

- Complex input(Scanner methods), Conditional loops (while)

Challenge

The game "Rock, Paper, Scissors" is frequently used by two players to determine a winner. Each player has three items: paper (P) , scissors (S) and a rock (R). At the count of three, each player shows one of them. The winner of a round is determined by the rules:

- paper covers rock
- scissors cut paper
- rock breaks scissors

If the two players show the same item then it is a draw, and neither player wins the round.
For this particular program, the overall winner is the first player to win two rounds.

The input will contain pairs of letters separated by single spaces. Each pair of letters make up a single round between two players. The letters are either "P", "S", or "R" (upper case), representing paper, scissors and rock, respectively. The players names are "PLAYER ONE" and "PLAYER TWO". The first letter of the pair represents PLAYER ONE's item and second letter of the pair represents PLAYER TWO's item. There will be enough pairs of letters (or more than enough), on each line, to determine a winner of two rounds (although there may be more than that). The length of each line will not exceed 255 characters.

The program should loop until a Q is entered at the start of a game.

The output will contain the overall winner's name (PLAYER ONE or PLAYER TWO) and the number of draws that occur before the winner is determined, separated by a single space.

Sample Input

```
PR PS RR SR
PP RR SP SP SR PS
RP SR
Q
```

Sample Output

```
PLAYER TWO 1
PLAYER ONE 2
PLAYER TWO 0
```

Level 2: Now I Know my ABC’s

New Concepts

- for loops, nested loops, character manipulation, while loops

Challenge

You are to determine the frequency of each letter of the alphabet in a line of characters.

- Your program should continue accepting input until the input is ‘stop’.
- The lines will not be more than 255 characters
- The output will display the frequency of the letters of the alphabet, in alphabetical order, from the corresponding lines of the input file.
- Display only those letters that occurred at least once.
- Separate the letter (upper case) from the frequency with a hyphen.

Sample Execution

```
Input: This sentence has 4 t’s in it.
Output: A-1:C-1:E-3:H-2:I-3:N-3:S-4:T-4
Input: And this one only has one.
Output: A-2:D-1:E-2:H-2:I-1:L-1:N-4:O-3:S-2:T-1:Y-1
Input: stop
```

Level 3 - Basketball Statistics

New Concepts

- Complexity, Conditionals, Math

Challenge

Ms. Toms, who coaches the senior girls' basketball team, wishes to know the foul shot percentage of her top five shooters. She needs to know this so she can send her best foul shooter to the line when her team shoots a technical foul.

The foul shot percentage is calculated by dividing the *number of shots made* by the *number of shots taken* and then *multiplying by 100.*

The input will contain four sets of data typed by the user.. Each set contains three lines, the player's name on the first line, the *number of shots taken* on the second line and the *number of shots made* on the third line.

The output will contain five lines of data. The first four will corresponding to each set of data in the input file. Each line will contain the player's name, a hyphen and the foul shot percentage rounded to two decimal places. The fifth line will print who has the highest percentage of shots made.

Sample Input

```
KATIE
24
12
JULIA
25
13
HILLARY
34
19
AMANDA
14
0
```

Sample Output

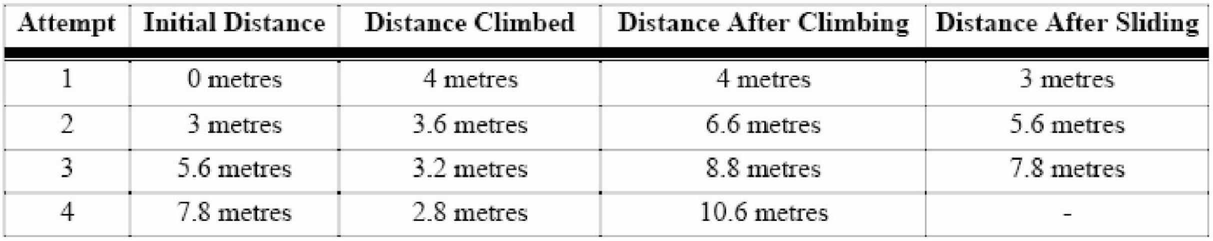
```
KATIE-50.00
JULIA-52.00
HILLARY-55.88
AMANDA-0.00
```

HILLARY is the best player.

Level 4 – Nicholas Loves Tobogganing

Nicholas is at the bottom of a 10 metre snow-covered hill and wants to climb to the top. He can climb 4 metres before he falls and slides down 1 metre. Nicholas has a fatigue factor of 10%, which means that on each successive try, he climbs 10% X 4 = 0.4 metres less than he did on the previous attempt. The distance lost to fatigue is always **10% of the first attempt's** climbing distance. On which attempt does Nicholas reach the top of the hill, i.e., on which attempt does Nicholas exceed the distance up the hill?

As you can see from the following table, Nicholas would reach the top of the hill on his fourth attempt.



Attempt	Initial Distance	Distance Climbed	Distance After Climbing	Distance After Sliding
1	0 metres	4 metres	4 metres	3 metres
2	3 metres	3.6 metres	6.6 metres	5.6 metres
3	5.6 metres	3.2 metres	8.8 metres	7.8 metres
4	7.8 metres	2.8 metres	10.6 metres	-

Your job is to solve this problem in general. Depending on the parameters of the problem, Nicholas will eventually either reach the top of the hill or slide back to the bottom of the hill. (In other words, Nicholas' distance climbed will exceed the distance of the hill or it will become negative.) You must find out which happens first and on which attempt.

The input will contain four integers *D*, *C*, *S*, and *F*, separated by a single space. All four numbers will be between 1 and 100, inclusive. *D* is the distance to the top of the hill in metres, *C* is the distance in metres that Nicholas can climb during his first attempt, *S* is the distance in metres that he slides down after he falls, and *F* is the fatigue factor expressed as a percentage.

Notes:

- Nicholas *never* climbs a negative distance. If the fatigue factor drops Nicholas' climbing distance to below zero, he does not climb at all on that attempt.
- Regardless of how far he has climbed, he always slides *D* feet after a fall.
- It is possible for Nicholas to slide back to the bottom of the hill (zero), but not "below" the hill (negative) (failure)

The output will contain a line indicating whether Nicholas succeeded (reached the top of the hill) or failed (slid back to the bottom) and on which attempt. Format the output ***exactly*** as shown in the sample.

Sample Input (only three cases given)

```
10 4 1 10
25 10 1 50
45 5 3 14
```

Sample Output

```
SUCCESS ON ATTEMPT 4
FAILURE ON ATTEMPT 16
FAILURE ON ATTEMPT 7
```

Level 5: Smile with similes

New Concepts

- Arrays, nested loops

Challenge

A simile is a combination of an adjective and noun that produces a phrase such as "Easy as pie" or "Cold as ice".

Write a program to input *n* adjectives (1 _ *n* _ 5) and *m* nouns (1 _ *m* _ 5), and print out all possible similes. The first two lines of input will provide the values of *n* and *m*, in that order followed, one per line, by *n* adjectives and *m* nouns. Your program may output the similes in any order.

Sample Input:

```
3
2
Easy
Smart
Soft
pie
rock
```

Sample output:

```
Easy as pie
Easy as rock
Smart as pie
Smart as rock
Soft as pie
Soft as rock
```