

Task 3

Preamble

In the land of Skibidi, a fierce conflict has erupted, pitting the Skibidi Toilets, sentient commodes with human visages, against a coalition of technologically augmented warriors—the Cameramen, Speakermen, and the recently allied TV-headed humanoids.

As dawn breaks over the battlegrounds, a sinister echo resounds through the fog:

"Bop."

This single, haunting sound marks the advance of the Skibidi Toilets. Led by the fearsome G-Toilet, these creatures move with eerie synchronicity, their intentions as cold and unforgiving as their porcelain faces. With each "bop," they inch closer to domination, their chilling chant striking fear into the hearts of all who hear it.

"Bop bop."

In response, the alliance of Cameramen and Speakermen rallies. Their call, a sharp and rhythmic double tap, orchestrates their movements across the battlefield. United by their common foe, they execute intricate manoeuvres designed to protect their territory and thwart the Skibidi advance. Their sound is one of determination, a beacon of resistance in the darkening world.

"Bop bop bop."

Amidst the chaos, a deeper tale of betrayal unfolds. The Speaker Titan, once a towering ally, now marches against its brethren. Infected by a nefarious Toilet parasite, its triple cry is a dire warning of its turned allegiance. This sound of treachery adds a layer of desperation to the conflict, as the alliance struggles to contain this new threat while also battling the relentless Skibidi Toilets.

Into this fray, the TV-headed humanoids emerge, their arrival marked by new strategies and an unpredictable element that could shift the balance of power. Their presence is a glimmer of hope, a chance to turn the tide against the overwhelming darkness.

This is not a battle for territory, but a fight for the soul of Skibidi itself. Will the eerie call of "bop" signify the end, or can the resonant "bop bop" and desperate "bop bop bop" rally the forces of good to reclaim their world? The fate of Skibidi hangs in the balance, with each sound echoing the hopes, fears, and valour of its inhabitants. The war is far from over, and its outcome uncertain, as each side seeks to outmanoeuvre the other in this bizarre and captivating conflict.

Task

Objective

This task involves simulating strategic confrontations between two factions: the Skibidis and the Alliance. Data for each faction, including character power levels and special abilities with specific arguments, are provided in external files.

Develop a Java program to read, parse and simulate battles using data from two distinct files. The program should calculate the outcome of the battle based on character power scores and parameterised special abilities, then output the battle results.

Input Specifications

Command-Line Arguments

Execute the program with paths to the faction data files as arguments:

```
java Skibidi alliance.txt skibidi.txt
```

File Input: Two separate files containing data for the Skibidi and the Alliance factions are required. The paths to these files are provided as command-line arguments, with the file for the Alliance always being given first and the Skibidi being given second.

File Format: Each file should follow this structure:

```
<Character1> <Power1>
<Character2> <Power2>
...
<Ability1> <Parameter1>
<Ability2> <Parameter2>
...
```

Characters are listed with their corresponding power scores, followed by any special abilities that influence the faction's overall battle capabilities.

Example File Content for Alliance (`alliance.txt`):

```
Cameraman 120
Speakerman 130
TV-Head 140
PowerSurge 50
BopShield 20
```

Example File Content for Skibidi (skibidi.txt):

```
G-Toilet 150  
ZapTrap 10  
BopShield 20  
TitanBoost 2
```

List of Special Abilities:

- **PowerSurge** <percentage> : Increases the power score of the first character in the file by the specified percentage.
- **BopShield** <percentage> : Increases total power by the specified percentage for all characters listed before this ability.
- **EnergyBoost** <points> : Adds the specified number of points to each character's score.
- **ZapTrap** <percentage> : Reduces the opponent's total power score by the specified percentage.
- **TitanBoost** <factor> : Multiplies the power of the last character listed before this ability by the given factor.

Output Specifications

Result: After processing the battle data, output the result.

```
Battle result: <Winner>
```

Winner Options:

- "Skibidi wins"
- "Alliance wins"
- "Draw" if the scores are equal after all calculations.

Error Handling

- **File Not Found:** If one of the input files cannot be found (you can assume at least one file will always be found):

```
Error: File '<filename>' not found. Please check the path and try again.
```

- **Invalid Ability Parameter:** If a parameter for an ability is not valid (non-numeric):

```
Error: Invalid parameter for '<ability>' in line '<line_content>'. Line skipped.
```

- **Unknown Ability:** If an ability listed does not match any predefined abilities:

```
Warning: Unknown ability '<ability>' detected in '<filename>'. Ignored.
```

- **Empty File:** If a file contains no valid data (only malformed lines or empty):

```
Error: No valid data found in '<filename>'. Battle cannot proceed.
```

Example Execution

Command Execution:

```
java Skibidi alliance.txt skibidi.txt
```

File Contents:

- **alliance.txt:**

```
Cameraman 120  
Speakerman 130  
TV-Head 140  
PowerSurge 50  
BopShield 20  
EnergyBoost 30
```

- **skibidi.txt:**

```
G-Toilet 150  
ZapTrap 10  
BopShield 20  
TitanBoost 2
```

Output:

```
Battle result: Alliance wins
```

Steps to Get Started

- **Initialisation:** Verify the presence and accessibility of input files.
- **Data Parsing:** Read and validate each file; handle format errors and skip problematic lines.

- **Battle Simulation:** Calculate the total scores for each faction, applying abilities and checking for numeric issues.
- **Determine Outcome:** Compare the final scores to decide the winner or identify a draw.
- **Output Results:** Display the battle outcome, ensuring all errors and warnings are clearly communicated during the process.

Frequently Asked Questions

What about errors that could be made that aren't described in the specifications?

Any possible erroneous circumstances that a user may encounter that haven't been described in the specifications do not need to be considered and will not be tested in the test cases.

Why is this task about Skibidi?

[#642](#)