

Entry Exam – Lord of The Rings

1.	Counting Orcs	1
2.	Council of Elrond Votes	1
3.	Elvish Code Cipher	2
4.	Orthanc Archives Search	2
5.	Legendary Duels	3
6.	Rohan Cavalry Formation	3

May your problem-solving skills be as keen as Andúril and may the wisdom of Gandalf guide your code!

1. Counting Orcs

e:
blvd.
m:

Aragorn is training his sword skills by fighting Orcs in Mirkwood forest. Given that he fights n Orcs in the first hour and increases his count by m every subsequent hour, determine the total number of Orcs he will have fought after h hours.

- **Input:** Three integers:
 - n represents the initial count
 - m represents the increase of the count.
 - h represents the number of hours.
- **Constraints:** n increases by m every hour.

Input	Output
5, 2, 3	21
7, 1, 2	15
10, -1, 1	10
8, -2, 3	18

...corrupted race of elves, either bred that way by the Dark Lord Morgoth...

2. Council of Elrond Votes

During the Council of Elrond, the members are voting on whether to destroy the One Ring. Members can vote "Yes", "No", or "Abstain". Given an array of votes, determine the decision of the Council – Yes, No, Abstain, or Tie.

- **Input:** An array of strings (votes).



Input	Output
["Yes", "No", "Yes", "Abstain", "Yes"]	Yes
["No", "No", "Yes"]	No
["Yes", "No"]	Tie
["Abstain"]	Abstain
["No", "No", "Abstain", "Abstain", "Yes", "Yes"]	Tie
["No", "Abstain", "Abstain"]	No

"You shall be the Fellowship of the Ring!" – Elrond

3. Elvish Code Cipher

The Elves have a secret code that they use to send encrypted messages. The code replaces each letter in a message with the letter that is n places before(smaller) in the alphabet. Write a program that deciphers the message.

e:
blvd.
m:

- Input: A string s ($1 \leq |s| \leq 100$) representing the encrypted message and an integer n ($1 \leq n \leq 25$) representing the shift.

Input/Output:

Input	Output
'Uifsf!jt!b!tfdsfu', 1	There is a secret!
'Wkurz#lw#lq#wkh#iluh\$', 3	Throw it in the fire!
"Fwfltws", 5	Aragorn

"The day has come! Behold, people of the Eldar and Fathers of Men, the day has come!"

4. Orthanc Archives Search

The Isengard tower - Orthanc records contain a vast array of data. Write a program that finds the first occurrence and the last occurrence of a specific record in the archive. If the record is missing, print "Record not found".

- Input: An array of strings representing the records and a string representing the search term.

Input/Output:

Input	Output
["Frodo", "Sam", "Frodo", "Merry"], "Frodo"	First Occurrence: 0 Last Occurrence: 2

["Pippin", "Merry", "Sam", "Pippin"], "Pippin"	First Occurrence: 0 Last Occurrence: 3
["Aragorn", "Legolas", "Gimli"], "Boromir"	Record not found
["Gandalf", "Sam", "Aragorn", "Frodo"], "Frodo"	First Occurrence: 3 Last Occurrence: 3

...Gandalf the Grey caught like a fly in a spider's treacherous web! ...

5. Legendary Duels

During a sword fight, every strike has a counterstrike.

- If an opponent uses a strike technique represented by (, the hero counters with a **technique** represented by).
- If an opponent uses a power strike technique represented by !, the hero counters with another power strike technique !.
- If an opponent uses the **magic** represented by {, the hero counters also using the **magic** }.

e:
blvd.
m:

If a sequence of strikes and counterstrikes is balanced in the right order, the fight is considered legendary. Determine if a given sequence is legendary.

- Input: A string sequence of strikes and counterstrikes.

Input/Output:

Input	Output
"()()()())"	Legendary
"((!!)(({!!})))"	Legendary
"((())"	Not Legendary
"{!}!"	Not Legendary
"({(!!})"	Not Legendary

"It was about a quarter mile below the fortress, and stretched a mile or more across the Deeping-coomb..."

6. Rohan Cavalry Formation

The Rohan cavalry is arranged in a specific formation that determines their charge pattern. The cavalry undergoes various operations and maintains its formation throughout the battle. Your program needs to execute a series of commands and return the new formation after each valid command.

Description:

- Input: An initial array of integers representing rider IDs and a list of commands.

Commands:

1. **"destroy [index]"**: The rider with the given index is destroyed and removed from the formation.



2. **"swap [index1] [index2]":** Swap the positions of two riders with the given indices.
3. **"add [ID]":** A new rider joins the formation, positioning itself at the end.
4. **"insert [ID] [index]":** Insert a new rider with the given ID at a specific position.
5. **"center":** Display the rider in the center of the formation. If there's an even number of riders, display the middle two.

Note: Ensure the commands handle edge cases, e.g., destroying a non-existent rider or inserting at a position out of bounds should take no action.

Input	Output
[1, 2, 3, 4, 5], ["destroy 3", "swap 0 1", "add 6", "center"]	1 2 3 5 2 1 3 5 2 1 3 5 6 3
[1, 2, 3, 4, 5], ["add 6", "swap 0 5", "swap 1 4", "swap 2 3", "swap 2 100", "swap 2 2", "center"]	1 2 3 4 5 6 6 2 3 4 5 1 6 5 3 4 2 1 6 5 4 3 2 1 4 3
[1, 2], ["insert 3 2", "center", "destroy 1", "destroy 2", "center"]	1 2 3 2 1 3 1 3

e:
blvd.
m:

*"They are proud and willful, but they are true-hearted,
generous in thought and in deed; bold but not cruel..."*