Foundation Exam

[1. Counting Womp Rats 1](#_Toc146784341)

[2. Galactic Senators' Votes 1](#_Toc146784342)

[3. Sith Code Cipher 2](#_Toc146784343)

[4. Jedi Archives Search 2](#_Toc146784344)

[5. Balanced Lightsaber Duels 3](#_Toc146784345)

[6. Starfighter Formation 3](#_Toc146784346)

May your problem-solving skills be as sharp as a lightsaber, and may the Force guide your code!

# Counting Womp Rats

Luke Skywalker is practicing his blaster skills by targeting Womp Rats on Tatooine. Given that he targets **n** Womp Rats in the first hour and increases his count by **m** every subsequent hour, determine the total number of Womp Rats he will have targeted 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**:

|  |  |
| --- | --- |
| Input | Output |
| 5, 2, 3 | 24 |
| 7, 1, 2 | 15 |
| 10, -1, 1 | 10 |
| 8, -2, 3 | 18 |

…the foul creature could grow to be as big as a man and could carry off children and full-grown Jawas…

# Galactic Senators' Votes

The Galactic Senate is voting on a new law. Senators can vote "Yes", "No", or "Abstain". Given an array of votes, determine the majority vote – Yes, No, Abstain or Tie.

* Input: An array of strings (votes).

**Input/Output**:

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

"There is no civility, only politics."

―Sheev Palpatine

# Sith Code Cipher

The Sith 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 ahead of it in the alphabet. Write a program that deciphers the message.

* Input: A string **s** (1 ≤ |s| ≤ 100) representing the encrypted message and an integer **n** (1 ≤ n ≤ 25) representing the shift.

**Input/Output**:

|  |  |
| --- | --- |
| Input | Output |
| "Uifsf jt b tdifsu", 1 | There is a secret |
| "Dagobah", 3 | Xyletdu |
| "R2-D2", 5 | W7-I7 |

"They hoped to fill me with fear. But fear leads to anger. Anger leads to hate. And hate…leads to power."

# Jedi Archives Search

The Jedi Archives contains 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 |
| ["Yoda", "Obi-Wan", "Yoda", "Luke"], "Yoda" | First Occurrence: 0  Last Occurrence: 2 |
| ["Han", "Leia", "Chewbacca", "Han"], "Han" | First Occurrence: 0  Last Occurrence: 3 |
| ["Lando", "Vader", "Palpatine"], "Maul" | Record not found |
| ["Yoda", "Obi-Wan", "Anakin", "Luke"], "Luke" | First Occurrence: 3  Last Occurrence: 3 |

…Files, or profiles, were kept of each member, whether alive, dead, or if they left the order…

# Balanced Lightsaber Duels

During a lightsaber duel, every strike has a counterstrike.

* If a Jedi uses a strike technique represented by **(**, the Sith counters with a technique represented by **)**.
* If a Jedi uses a power strike technique represented by **!**, the Sith counters with another power strike technique **!**.
* If a Jedi uses the force represented by **{,** the Sith counters also using the Force **}**.

If a sequence of strikes and counterstrikes is balanced, the duel 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 |

"An elegant weapon for a more civilized age."

―Obi-Wan Kenobi

# Starfighter Formation

Starfighters in a squadron are arranged in a specific formation that determines their attack pattern. The squadron 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.

**Description:**

* Input: An initial array of integers representing starfighter IDs and a list of commands.

**Commands**:

1. **"destroy [index]"**: The starfighter with the given index is destroyed and removed from the formation.
2. **"swap [index1] [index2]"**: Swap the positions of two fighters with the given indices.
3. **"add [ID]"**: A new fighter joins the formation, positioning itself at the end.
4. **"insert [ID] [index]"**: Insert a new fighter with the given ID at a specific position.
5. **"center"**: Display the fighter in the center of the formation. If there's an even number of fighters, display the middle two.

Note: Ensure the commands handle edge cases, e.g., destroying a non-existent fighter 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 |

"We'll have to destroy them ship to ship. Get the crews to their fighters."

―Darth Vader