Program 1:

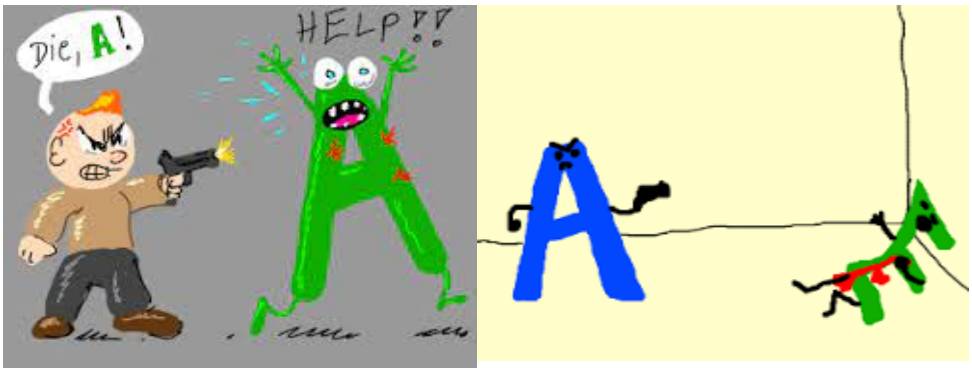
Your computer science teacher is trying to analyze the performance of the class in the previous exam. He has a class called Performance, which contains the marks of 10 students in the class. These are not sorted. He wants to find out four quantities.

Mode - the most frequently occurring marks in the class. If two or more marks occur equally frequently then the highest of these marks is the mode.
Mode frequency – frequency at mode.
Highest Mark - Highest mark in the class
Least Mark – Least mark in the class
You can make the following assumptions.
The class has 10 students
The maximum marks anyone get are 100 and the minimum is 0.
All students marks are whole numbers.
Important: You are not allowed to sort the marks.
Some of the members of **Performance** class are given below.
Data member :
mark[] -> an integer to store the marks of 60 students.
Member functions

Performance()	: constructor
void readMarks()	: for reading the marks into the array
int highestMark()	: To return the highest mark scored in the class
int leastMark()	: To return the least mark scored in the class
int getMode()	: for returning the mode
int getFreqAtMode()	: for returning the frequency at mode
void display()	: To display the result

Specify the class **Performance** and implement it in Java.

Program 2:



2)Imagine you really enjoy playing games, especially "**CALL OF DUTY**," and you're pretty good at using Java. Now, someone gives you a cool task: make a game called "**Alphabet War**."

This game is all about a playful fight between letters, split into two teams—the left-side letters and the right-side letters.

The left-side letters act like superheroes, each having their own strengths:
- 'w' is super strong (strength 4)
- 'p' is quite strong (strength 3)
- 'b' is okay strong (strength 2)
- 's' is kinda strong (strength 1)

On the other side, the right-side letters also have their own strengths:
- 'm' is super strong (strength 4)
- 'q' is quite strong (strength 3)
- 'd' is okay strong (strength 2)
- 'z' is kinda strong (strength 1)

Your job is to set up some rules for this friendly letter battle. If the left-side letters win, you say "Left side wins!" If the right-side letters win, you say "Right side wins!" If it's a tie or something else, you say "Let's fight again!"

To make things fun in Java, you create a class called `AlphabetWarGame`. This class helps you figure out who wins using different rules for different scenarios. You can even change the strengths of the letters if you want.

For example, you might create this class in different ways using **constructor overloading:**
- One version where the strengths are set by default.
- Another version where you can customize the strengths.

And, you can figure out who wins in different ways using **method overloading:**
- One method where you pass in just one word.
- Another method where you pass in separate left and right words.

Following are few sample testcases for you to implement:

```
AlphabetWar("z") => Right side wins!
AlphabetWar("zdqmwpbs")=> Let's fight again!
AlphabetWar("wwwwwwz" )=> Left side wins!
```

It's like making your own world of letter superheroes in Java, having a blast experimenting with different strengths and battle scenarios!