



Data Structures and Algorithms

Unit 2.1 – Stack ADT

Unit 2.2 – Queue ADT

Problem 1 – Balanced arithmetic expression.

Implement a Python class to guess if the delimiters `()`, `{}`, `[]` in an arithmetic expression (e.j. `[(5+x)-(y+z)]`) are balanced.

- Correct expression example: `()(){}([()])`
- Incorrect expression example: `{[]}()`

Use a stack to implement your solution. Consider the following hints:

- If an opening symbol is found `[, {, (` it must be pushed.
- If a closing symbol is found `], },)` the element at the top of the stack must be queried. If both symbols belong to the same type, the element must be removed.
- The arithmetic expression is balanced if at the end of the process the stack is empty.

Problem 2 – Josephus problem.

In the Jewish revolt against Rome, Josephus and 39 of his mates were holding out against the Romans in a cave. With defeat imminent, they resolved that they would rather die than be slaves to the Romans. They decided to arrange themselves in a circle. One man was designated as number one, and they proceeded clockwise killing every seventh man (step). Josephus was among other things an accomplished mathematician; so he instantly figured out where he ought to sit in order to be the last to go. But when the time came, instead of killing himself he joined the Roman side.

Implement a method to find out what position Josephus should sit in order to not be killed. The solution should generalize for any number of Jewish soldiers and any step. The solution should use a queue of integers (each soldier is represented with a number from 1 to n).

In the following video, you can find a nice explanation of this problem.

<https://www.youtube.com/watch?v=uCsD3ZGzMgE>