Inchworm

Problem Statement

The inchworm is a creature of regular habits. She inches forward some distance along the branch of a tree, then stops to rest. If she has stopped at a leaf, she makes a meal of it. Then she inches forward the same distance as before and repeats this routine until she has reached or passed the end of the branch.

Consider an inchworm traveling the length of a branch whose leaves are spaced at uniform intervals. Depending on the distance between her resting points, the inchworm may or may not be able to eat all the leaves. There is always a leaf at the beginning of the branch, which is where the inchworm rests before setting out on her journey.

You are given three int values that specify, in inches: the length of the branch; the distance travelled by the inchworm between rests; and the distance between each consecutive pair of leaves. Given that the inchworm only eats at rest, calculate the number of leaves she will consume.

Definition

Class: Inchworm
Method: lunchtime
Parameters: int, int, int
Returns: int
Method signature: int lunchtime(int branch, int rest, int leaf)
(be sure your method is public)

Notes

- The inchworm starts by gobbling up the leaf at the beginning of the branch.
- If there is a leaf at the end of the branch, the inchworm eats it only if it falls at a resting point.

Constraints

- branch is between 1 and 1000000 (one million), inclusive
- rest is between 1 and 1000, inclusive
- leaf is between 1 and 1000, inclusive

Examples

0) 11
2
4
Returns: 3
Leaves grow 0, 4, and 8 inches from the beginning of the branch. The inchworm eats them all.

1) 12
6
4
Returns: 2
The inchworm misses the leaves growing at distances 4 and 8, but eats those at 0 and 12.

2) 20
3
7
Returns: 1
The inchworm eats only the leaf at the outset.

3) 21
7
3
Returns: 2
The inchworm eats the leaves at the beginning and end of the branch.

4) 15
16
5
Returns: 1
The inchworm eats only the leaf at the outset.

5) 1000
3
7
Returns: 48

6) 1000
7
3
Returns: 48

ThrowTheBall

Problem Statement

There are N friends sitting in a circle, numbered clockwise in increasing order from 1 to N. At the beginning of the game, player 1 receives a ball. The players take turns passing the ball to each other. At the beginning of the game and before each next pass the following actions are performed. If the player with the ball has already received the ball M times, the game is over. Otherwise, if the player has received the ball p times, he'll pass the ball directly to the person L places to his left if p is even, or L places to his right if p is odd (see examples for clarification). Given N, M and L, return the number of times that the ball is passed.

Definition

Class: ThrowTheBall
Method: timesThrown
Parameters: int, int, int
Returns: int
Method signature: int timesThrown(int N, int M, int L)
(be sure your method is public)

Constraints

- N will be between 3 and 50, inclusive.
- M will be between 1 and 50, inclusive.
- L will be between 1 and N-1, inclusive.

Examples

0) 5
3
2
Returns: 10
First, player 1 gets the ball. Since he has held the ball 1 time, he passes the ball to player 4, who is two places to his right. This is player 4's first time holding the ball, so he gives it to player 2, who passes it to player 5. Player 5 then passes the ball to player 3, who passes it back to player 1. Since player 1 has now held the ball 2 times, he passes it to player 3, who passes it to player 5, who then passes the ball to player 2. Finally, player 2 passes the ball to player 4, who then passes it to player 1. Player 1 has now held the ball 3 times, and the game ends.

1) 4
1
3
Returns: 0
Here, the ball is never passed.

2) 10
3
5
Returns: 4

3) 15
4
9
Returns: 15

Reppity

Problem Statement

Given a String, input, with up to 50 characters, find the length of the longest substring that appears at least twice (non-overlapping) in input. If no substring appears twice, non-overlapping, return 0.

Strings are case sensitive, and only upper case letters and lower case letters are allowed in input. For example, in the string "ABCDEXXXYYYZZZABCDEZZZYYYXXX" the longest substring which appears at least twice is "ABCDE". These two substrings do not overlap so you would return 5.

Definition

Class: Reppity
Method: longestRep
Parameters: String
Returns: int
Method signature: int longestRep(String input)
(be sure your method is public)

Notes

- We are looking for subSTRINGS not subSEQUENCES. All of the elements of a substring appear consecutively in the original string. In other words, the substring can be formed from the original string by deleting zero or more characters from the beginning and deleting zero or more characters from the end, but NO deletions from the middle are allowed.

Constraints

- input will contain between 1 and 50 characters inclusive.
- Each character in input will be between 'a' and 'z' or between 'A' and 'Z' inclusive.

Examples

0) "ABCDEXXXYYYZZZABCDEZZZYYYXXX"
Returns: 5
The example from above.

1) "abcdabcdabcdabCD"
Returns: 6
"abcdab"+"cd"+"abcdab"+"CD"

2) "abcdefghijklmnopqrstuvwxyabcdefghijklmnopqrstuvwxy"
Returns: 25

3) "againANDagainANDagainANDagainANDagainANDagain"
Returns: 21

4) "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWX"
Returns: 0