

A *word machine* is a system that performs a sequence of simple operations on a stack of integers. Initially the stack is empty. The sequence of operations is given as a string. Operations are separated by single spaces. The following operations may be specified:

- an integer X (from 0 to $2^{20} - 1$): the machine pushes X onto the stack;
- "POP": the machine removes the topmost number from the stack;
- "DUP": the machine pushes a duplicate of the topmost number onto the stack;
- "+": the machine pops the two topmost elements from the stack, adds them together and pushes the sum onto the stack;
- "-": the machine pops the two topmost elements from the stack, subtracts the second one from the first (topmost) one and pushes the difference onto the stack.

After processing all the operations, the machine returns the topmost value from the stack.

The machine processes 20-bit unsigned integers (numbers from 0 to $2^{20} - 1$). An overflow in addition or underflow in subtraction causes an error. The machine also reports an error when it tries to perform an operation that expects more numbers on the stack than the stack actually contains. Also, if, after performing all the operations, the stack is empty, the machine reports an error.

For example, given a string "4 5 6 - 7 +", the machine performs the following operations:

operation	comment	stack
"4"	push 4	[empty]
"5"	push 5	4
"6"	push 6	4, 5
"-"	subtract 5 from 6	4, 5, 6
"7"	push 7	4, 1
"+"	add 1 and 7	4, 1, 7
		4, 8

Finally, the machine will return 8.

Given a string "13 DUP 4 POP 5 DUP + DUP + -", the machine performs the following operations:

operation	comment	stack
"13"	push 13	[empty]
"DUP"	duplicate 13	13
"4"	push 4	13, 13
"POP"	pop 4	13, 13, 4
"5"	push 5	13, 13
"DUP"	duplicate 5	13, 13, 5
"+"	add 5 and 5	13, 13, 5, 5
		13, 13, 10

"DUP"	duplicate 10	13, 13, 10, 10
"+"	add 10 and 10	13, 13, 20
"-"	subtract 13 from 20	13, 7

Finally, the machine will return 7.

Given a string "5 6 + -", the machine reports an error. After the addition, there is only one number on the stack and the subtraction operation expects two.

Given a string "3 DUP 5 - -", the machine reports an error. The second subtraction yields a negative result.

Write a function:

```
def solution(S)
```

that, given a string S containing a sequence of operations for the word machine, returns the result the machine would return after processing the operations. The function should return −1 if the machine would report an error while processing the operations.

Examples:

1. Given S = "4 5 6 - 7 +", the function should return 8, as explained above.
2. Given S = "13 DUP 4 POP 5 DUP + DUP + -" the function should return 7.
3. Given S = "5 6 + -", the function should return −1.
4. Given S = "3 DUP 5 - -", the function should return −1.
5. Given S = "1048575 DUP +", the function should return −1.

Assume that:

- the length of string S is within the range [0..2,000];
- S is a valid sequence of word machine operations.

In your solution, focus on correctness. The performance of your solution will not be the focus of the assessment.

Copyright 2009–2024 by Codility Limited. All Rights Reserved. Unauthorized copying, publication or disclosure prohibited.