Problem F6404
Polish Notation

Polish notation is a form of notation for logic, arithmetic and algebra. The distinguishing feature is that it places operators to the left of their operands. What does this mean?

For example the term "1 + 2"(our common notation) will be represented by "+ 1 2" in Polish notation. Another example would be "(5 - 6) * 7" will be represented by "* - (5 6) 7 ". Since arithmetic operators are binary, the prefix representation is unambiguous, and bracket becomes unnecessary. So the expression can be simplified to "* - 5 6 7".

Let us try to think about Polish notation in a recursively. We could define a valid Polish notation to be the following.

1. A single number is a valid Polish notation expression.
2. An operator followed by two valid Polish notation expression will be valid.

By this definition, both "5" and "6" are single number, hence valid expressions. Thus "- 5 6" will be a valid expression as well. Since both "- 5 6" and "7", which is a single number, are valid expressions, therefore "* - 5 6 7" is a valid expression.

Given a valid Polish notation expression, compute the value of the expression.

Hint 1: The lecture notes have taught about calling global variable defined outside of the function in the function. However, to modify the value of the global variable, simply modify it as local variable will introduce an error. To modify the global variable inside the function, you may use a instruction inside the function: "global p" where "p" is a global variable. After that the function can modify the global variable "p" when it is called. You may try several codes to be familiar with this and proceed to the problem.

Hint 2: Set a global variable "p = 0" and read the expression as a list "s". "s[p]" can be the term you are currently processing. Then you may apply the recursive property of Polish notation and build a recursive function to return the value of a Polish notation.

Input

The only line of the input consist of a valid Polish notation expression. The expression consist of at least 1 and at most 15 space separated terms. Each term will be either one of the operator "+ - *", or an integer between 1 and 10 inclusive.

Output

Print a single integer on a line, the value of given expression.

Sample Input 1

```
+ 1 2
```

Sample Output 1

```
3
```

Sample Input 2

```
* - 5 6 7
```

Sample Output 2

```
-7
```

Sample Input 3

```
* + 3 9 + 10 6
```

Sample Output 3

```
192
```

Explanation of Sample Data 3

Let us try to read the term one by one, remember that we are looking at a valid expression.

1. The whole string is a valid expression, and the first term is an operator "*", so it should be followed by two valid expressions. We try to read the first valid expression, and we move to next term of the whole string.
    (a) We are currently at an operator "+", so it should be followed by two valid expressions, again we try to read the first valid expression, and we move to next term of the whole string.
        i. We are currently at an integer "3", which is a valid expression. So we have found the first valid expression for our previous "+" sign.
        ii. The next term is an integer "9", which is a valid expression. So we have found the second valid expression for our previous "+" sign.
        iii. Since all two terms for the "+" sign are founded. We can calculate the result of "+" sign to be "3+9=12", which will serve as the first valid expression for our very first "*" sign.

(b) We can now move the term after "9", which is "+" sign. This should be the start of the second valid expression of our very first "*" sign. And again we find it to be an operator. We try to read the first valid expression, and we move to next term of the whole string.

   i. We are currently at an integer "10", which is a valid expression. So we have found the first valid expression for our "+" sign.

   ii. The next term is an integer "6", which is a valid expression. So we have found the second valid expression for our "+" sign.

   iii. Since all two terms for the "+" sign are founded. We can calculate the result of "+" sign to be "10+6=16", which will serve as the second valid expression for our very first "*" sign.

(c) Finally we need to combine the two terms with multiplication, the result will be "12*16=192", which will be the value of whole expression.