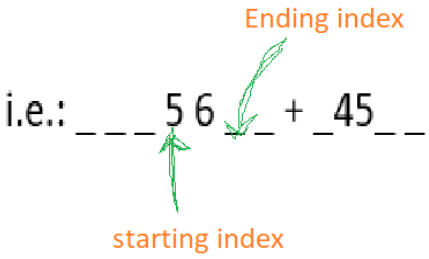
Problems occurs while solving this Problem Statement:

1) We don't know user give us a how many space
2) And our second problem is that we can't edit a user given string, so we have tried to find 2 **operands** and 1 **operator.**
3) If we have found illegal character like (a,b…z,&,^,$,@....) so we have to throw the error
   (For overcome this problem we have to check every character of the string and if we found the illegal character then at this character we have to throw error and break telse if we not found any illegal character then we have continue our logic)

4) We must manage the space at 4 places
   - Before first operand
   - After first operand
   - After operator or we can say that before second operand
   - After second operand
5) Manage the if conditions and variable in proper manner
6) If user not giving the space at one and more places out of these 4 places, so we must manage this kind of problems
7) We must create a program which are run at any kind of the situation
8) We must convert the string to integer for this we have use the sub string method and parse method for doing this.

```
i.e: First operand=Integer.parseInt(expr.substring(start_index, end_index))
```

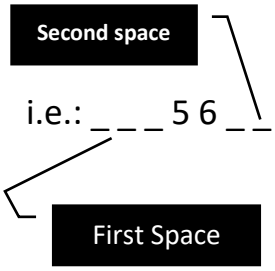
9) But problem is that what is the end index of the sub string



We expect that our **end_index** is the six, but it's not true because
In substring the **ending index** is the always plus one(**+1**) when where we have to end index as illustrate in image.

Solution:

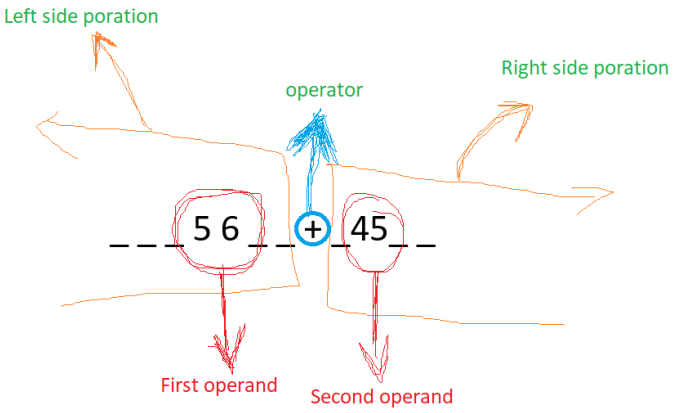
let suppose user give a one sting given below



i.e.: _ _ _ 5 6 _ _ + _45_ _ (in this **"_"** is indicated **white space****)

FYI: (as like first space and second space we have the third space and fourth space after operator.)

Before understand approach to this problem we must understand some terminology



Let Suppose index of operator is, **index = indx_o**

1) **Left side portion:** The left most part of the string before operator, from starting index (0) to (**indx_o** - 1).
2) **Right side portion:** The right most part of the string before operator, from starting (**indx_o** + 1) to length of string(**str.length()**).
3) **First operand:** operand which are found in the left side portion.
4) **Second operand:** operand which are found in the right-side portion.
5) **Operator:** operator which are used in arithmetic operation **(+, -, *, /).**