I have now finished modifying the code and it is in a compile free state.

I didn't expect it to be functional logically so it is perfect opportunity to understand.

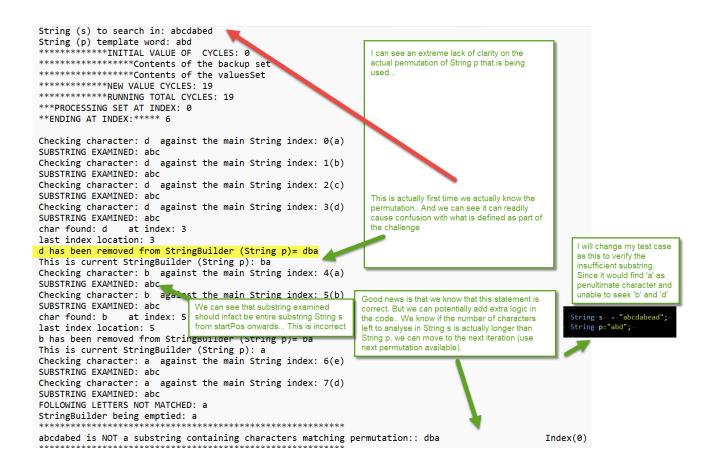
TEST CASE 1:

```
l expected it to find a match in
both of these scenarios

--String-s----"abcdabed";-
--String-p="abd";--
```

I have performed a search through all my screen outputs and there is no a single entry referring to this block of code:

My best technique is just to simply analyse the applicable startPos and follow logic of my code:



TEST CASE 2:

Implementing changes:

```
String s == "abcdabead"; -
String p="abd"; -
System.out.println("String (s) to search in: " ++ s); -
```

System.out.println("String (p) template permutation word: " + sb)

I will now run this test case and see if it reaches insufficient characters in String s

String (s) to search in: abcdabead

String (p) template permutation word: abd //I have taken this example since it matches template word, but could have easily taken adb

SUBSTRING EXAMINED: abc //I am not sure why its showing this as substring since we expected it to be every character from startPos. Even when scrolling down this screen output. I have expressed this in my code.. Since it's a massive screen output, I have remediated code and completed screen output again

TEST CASE 3:

String (s) to search in: abcdabead

String (p) template permutation word: abd

Checking character: a against the main String index: 0(a)

SUBSTRING EXAMINED: abcdabead

char found: a at index: 0 //This is correct

last index location: 0

a has been removed from StringBuilder (String p)= abd //This is correct

This is current StringBuilder (String p): bd

Checking character: b against the main String index: 1(b)

SUBSTRING EXAMINED: abcdabead

char found: b at index: 1 //This is correct

last index location: 1

b has been removed from StringBuilder (String p) = bd //This is correct

This is current StringBuilder (String p): d

Checking character: d against the main String index: 2(c)

SUBSTRING EXAMINED: abcdabead

Checking character: d against the main String index: 3(d)

SUBSTRING EXAMINED: abcdabead

char found: d at index: 3 //This is correct

last index location: 3

d has been removed from StringBuilder (String p)= d //This is correct

This is current StringBuilder (String p):

abc is a substring containing characters matching permutation: abd

Index(0) //There is a

mistake here, it should mention abcd and not abc

//We know that is infact since last index is exclusion, so need to increase the termination by 1

This is minimum window: 9 //more clarity if stated current minimum window

String p (all characters) found within: String s (index: 0 - 3) //This is correct

------STORING WINDOW RESULT: 3); // //this is incorrect.. We need to

instead perform subtraction

(lastIndexLocationStringS-startPos) + 1 due to zero indexing

ROW NUMBER: 1

I have managed to include all my above thought process into the code...

The only aspect not analysed and perhaps it's the most basic....

The biggest structural change was when running the first permutation template word and finding no matches in String s.

Also when there were insufficient characters left in String's to ensure that characters of the template word could be matched...