For once I have totally been deterred from a recursive approach.
I found this to be quite intense and it has taken 1-2 hours to draft a compile error free code.

As per usual, I will try to ascertain the issues in hand.

## TEST CASE : Executing the code

```
findAnagrams("cbaebabacd", "abc");
```

Even though I have had so much experience, at first glance the outputs seem a total mismatch to my objectives!

```
Welcome to Online IDE!! Happy Coding :)
Checking character: cin the main String index: 0(c
char found: a
This is current StringBuilder (String p): baebabacd
Checking character: ain the main String index: 0(c
char found: b
This is current StringBuilder (String p): bebabacd
Checking character: bin the main String index: 0(c
Checking character: bin the main String index: 0(c
Exception in thread "main"
java.lang.StringIndexOutOfBoundsException: String index out of range: 4
    at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:47)
    at java.base/java.lang.String.charAt(String.java:693)
    at Main.findAnagrams(Main.java:39)
    at Main.main(Main.java:15)
```

I think the first aspect is to output on the screen the initial String values.

It is still plagued with lots errors… It has found character a however the search was for letter c.
It has also displayed the StringBuilder to be totally incorrect.

```
Welcome to Online IDE!! Happy Coding :)
String (s) to search in: cbaebabacd
String (p) template word: abc
Checking character: c  in the main String index: 0(c)
char found: a
This is current StringBuilder (String p): baebabacd
Checking character: a  in the main String index: 0(c)
char found: b
This is current StringBuilder (String p): bebabacd
Checking character: b  in the main String index: 0(c)
Checking character: b  in the main String index: 0(c)
Exception in thread "main"
java.lang.StringIndexOutOfBoundsException: String index out of range: 4
   at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:47)
   at java.base/java.lang.String.charAt(String.java:693)
   at Main.findAnagrams(Main.java:43)
   at Main.main(Main.java:15)
```

I figured out I had overcomplicated the required screen output

```
41  if (s.substring(startPos,p.length()).indexOf(p.charAt(pos))!=-1)
42  {
43      //System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt(pos))));
44
45      System.out.println("char found: " + s.charAt(i));
```

So now I am in a position where my code is hanging indefinitely...

## TEST CASE: Hanging indefinitely

```
findAnagrams("cbaebabacd", "abc");
```

```
Welcome to Online IDE!! Happy Coding :)
String (s) to search in: cbaebabacd
String (p) template word: abc
Checking character: a  in the main String index: 0(c)
char found: c
This is current StringBuilder (String p): bc
Checking character: c  in the main String index: 0(c)
char found: c
This is current StringBuilder (String p): b
Checking character: b  in the main String index: 1(b)
char found: b
This is current StringBuilder (String p):
```

It can also be seen that the StringBuilder (String p) is incorrect.
It appears that I have removed the a as oppose to the c.
I expect this to be relatively straight forward error.

I have taken a look at my code again....
And it seems that since I am searching s.substring, it would be searching for character a across the entire part... So in effect there was a match, the dispute is character found, so there clearly is something wrong with the indexing and output to the screen only..

I have performed the correction to the below.

```
System.out.println("char found: ");/* + s.substring(startPos,p.length()).indexOf()*/

System.out.println(s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt(pos))));
```

So it can be seen that even my previous attempts were also wrong. I am beginning to think these mistakes should not happen. Perhaps for the first time (in a long time) am I feeling the intensity of an iterative challenge!

```
System.out.println("char found: ");/* + s.substring(startPos,p.length()).indexOf()*/

System.out.println(s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt(pos))));
```

```
System.out.println("char found: " + s.charAt(i));
```

## TEST CASE:  Running the code again with amendments above

```
findAnagrams("cbaebabacd", "abc");
```

I can see it looks perfectly fine BUT we can see that it has increased to pos=1

```
Welcome to Online IDE!! Happy Coding :)
String (s) to search in: cbaebabacd
String (p) template word: abc
value of pos: 0
Checking character: a  against the main String index: 0(c)
char found: a
This is current StringBuilder (String p): bc
value of pos: 1
Checking character: c  against the main String index: 0(c)
char found: c
This is current StringBuilder (String p): b
value of pos: 0
Checking character: b  against the main String index: 1(b)
char found: c
This is current StringBuilder (String p):
```

This was a part of my intended code logic

```
for (int i=startPos; i<s.length();i++)
{
    for (int pos=0; pos<sb.length(); pos++)
    {
```

But clearly performing this has resulted in comparison with
c against index 0(c)

I think there are few bits I need to address in my code first. I need to make it clear that it searching in the main String index from index 0 to  p.length()-1
Since this will avoid unnecessary confusion!

## TEST CASE: Incorporating large segment of System.out.println() to remediate onscreen messages

```
findAnagrams("cbaebabacd", "abc");
```

It can be seen now that for the first time, it is consistent with logic taking place...

```
37          {
38              System.out.println("value of pos: " + pos);
39              System.out.println("Checking character: " + sb.toString().charAt(pos) +
40              "  against the main String index: " + i + "("+s.charAt(i)+")"
41              + " TO index: " + (i+p.length()) +  "("+s.charAt((i + (p.length()-1)))+")" );
42
43              //if there is a match
44              if (s.substring(startPos,p.length()).indexOf(p.charAt(pos))!=-1)
45              {
```

Ln: 34, Col: 51

Stop | Share | Command Line Arguments

```
Welcome to Online IDE!! Happy Coding :)
String (s) to search in: cbaebabacd
String (p) template word: abc
value of pos: 0
Checking character: a  against the main String index: 0(c)TO index: 3(a)
char found: a
This is current StringBuilder (String p): bc
```

I will now continue to look at issue identified above..
We know moving pos forward is not suited to the logic when it finds a match.
Since the pos variable is used to index the StringBuilder which undergoes a character removal...
**Hence initialisation has to be set again to pos=0. So logic has to be introduced for this scenario.**

```
//if there is a match
if (s.substring(startPos,p.length()).indexOf(p.charAt(pos))!=-1)
{
    //System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.char

    System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt

    //it removes character from the StringBuilder(which holds String p)
    //at index which it found the match
    sb.deleteCharAt(pos);
    startPos++;
    System.out.println("This is current StringBuilder (String p): " + sb.toString());
    pos=0;
}
```

Although it initially felt like this was correct operation, this is not the route forward..
Since it would need to break out of the for loop to enforce this..
And then it will serve no purpose given that pos is initialised to 0 anyhow.

So I have included the following logic in key areas:

```
37              for (int pos=startPos; pos<sb.length(); pos++)
38              {
39                  if (hasCharFound)
40                  {
41                      pos=pos-1;
42                  }
```

```
if (s.substring(startPos,p.length()).indexOf(p.charAt(pos))!=-1)
{
    //System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).index

    System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf

    //it removes character from the StringBuilder(which holds String p)
    //at index which it found the match
    sb.deleteCharAt(pos);
    startPos++;
    System.out.println("This is current StringBuilder (String p): " + sb.toString());
    hasCharFound=true;
```

We also know that if there is no match, then its perfectly fine to start at pos=0. We know the code is currently geared towards this.

We also know that once it exits out the inner do while loop, it has finished processing String p.
The code will automatically loop and re-initialise pos=0

```
62          }
63        }while(startPos<=s.length());
64
```

## TEST CASE:  Executing the code again

```
findAnagrams("cbaebabacd", "abc");
```

There is also no looping in the code, which is a good area

```
Welcome to Online IDE!! Happy Coding :)
String (s) to search in: cbaebabacd
String (p) template word: abc
value of pos: 0
Checking character: a  against the main String index: 0(c) TO index: 3(a)
char found: a
This is current StringBuilder (String p): bc
value of pos: 0
Checking character: b  against the main String index: 0(c) TO index: 3(a)
char found: b
This is current StringBuilder (String p): c
REACH1
value of i: 0
value of startPos: 2
value of pos: -1
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: -1
   at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:47)
   at java.base/java.lang.String.charAt(String.java:693)
   at Main.findAnagrams(Main.java:44)
   at Main.main(Main.java:15)
```

However we can see that value of pos=-1
This has arisen due to performing pos=pos-1
Perhaps best approach is to simply set pos=0 as below.

```
            //this is going through the main String...
            //it is starting more inwards once it has processed the entire length of p
            for (int i=startPos; i<s.length();i++)
            {
                for (int pos=0; pos<sb.length(); pos++)
                {
                    if (hasCharFound)
                    {
                        pos=0;
                    }
```

## TEST CASE: Re-executed the code

```
findAnagrams("cbaebabacd", "abc");
```

It can be seen that code has not even left the first inner do while loop.

```
value of i: 9
value of startPos: 3
REACH1
value of i: 3
value of startPos: 3
REACH1
value of i: 4
value of startPos: 3
REACH1
value of i: 5
value of startPos: 3
REACH1
value of i: 6
value of startPos: 3
REACH1
value of i: 7
value of startPos: 3
REACH1
value of i: 8
value of startPos: 3
REACH1
value of i: 9
value of startPos: 3
REACH1
value of i: 3
value of startPos: 3
```

```
                        break;
CODE IS                 }
UP TO               }
HERE                System.out.println("REACH1");
                    System.out.println("value of i: " + i);
                    System.out.println("value of startPos: " + startPos);
                } //end of for loop    for (int i=startPos; i<s.length();i++)
            }while(startPos<=s.length());
                                        THE CODE HAS NOT
            System.out.println("REACH2"); REACHED HERE
```

It can be clearly seen that the while loop is incorrect.

Since we know the startPos will be incremented by 1 in event that a char is found or not found... This makes perfect sense.... Since we know moving one character in main String is required...

# TEST CASE:

```
findAnagrams("cbaebabacd", "abc");
```

```
Checking character: a  against the main String index: 0(c) TO index: 3(a)
char found: a
This is current StringBuilder (String p): bc
value of pos: 0
value of sb: bc
Checking character: b  against the main String index: 0(c) TO index: 3(a)
char found: b
This is current StringBuilder (String p): c
value of pos: 0
value of sb: c
Checking character: c  against the main String index: 0(c) TO index: 3(a)
char found: c
This is current StringBuilder (String p):
REACH1
value of i: 0
value of startPos: 3
value of pos: 0
value of sb:
Exception in thread "main"
java.lang.StringIndexOutOfBoundsException: String index out of range: 0
    at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:47)
    at java.base/java.lang.String.charAt(String.java:693)
    at Main.findAnagrams(Main.java:45)
    at Main.main(Main.java:15)
```

It can be seen that sb is empty, so that has to be part decision to exit do while loop

I have firstly adjusted the do while loop as below:

```
81
82          //I am changing the exit condition to this...
83          //So this effectively means that any areas of code above where it doesn't find an anagram, it should in practice
84          //remove all contents of the StringBuilder.
85          //And only once it leaves the while loop, it should aim to restore it again...
86          }while(!sb.toString().isEmpty()  /*startPos<=p.length()*/);
87
88          System.out.println("REACH2");
89          //at this point we know that it has found an anagram
90          //since all characters have been removed
91
92          if (sb.toString().isEmpty() && hasCharFound)
93          {
94              System.out.println(s.substring(startPos,p.length()) + " is an anagram of: " + p);
95          }
```

And now this effectively means that before where it did not find a match, I restored the original StringBuilder (within the while loop)...
However I need to ensure that if there is no match, I clear the StringBuilder to the entirety otherwise it will not leave the inner do while loop.

```
62                  }
63                  //if there is no match found, no need to iterate inner for loop any further
64                  //but the outer loop is still relevant since it might find anagram starting from one position
65                  //further in the main string
66
67                  else
68                  {
69                      System.out.println("NO MATCH FOUND");
70                      pos=p.length();
71                      startPos++;
72                      sb.delete(0,sb.length());
73                      hasCharFound=false;
74                      break;
75                  }
76              }
77              System.out.println("REACH1");
78              System.out.println("value of i: " + i);
79              System.out.println("value of startPos: " + startPos);
```

I have performed this operation

I will execute the code again now...

## TEST CASE:

```
findAnagrams("cbaebabacd", "abc");
```

It appears that it has completed the full cycle, it has shown that it has found match with a, b and c

```
Welcome to Online IDE!! Happy Coding :)
String (s) to search in: cbaebabacd
String (p) template word: abc
value of pos: 0
value of sb: abc
Checking character: a  against the main String index: 0(c) TO index: 3(a)
char found: a
This is current StringBuilder (String p): bc
value of pos: 0
value of sb: bc
Checking character: b  against the main String index: 0(c) TO index: 3(a)
char found: b
This is current StringBuilder (String p): c
value of pos: 0
value of sb: c
Checking character: c  against the main String index: 0(c) TO index: 3(a)
char found: c
This is current StringBuilder (String p):
REACH1
value of i: 0
value of startPos: 3
value of pos: 0
value of sb:
Exception in thread "main" java.lang.StringIndexOutOfBoundsException: String index out of range: 0
    at java.base/java.lang.StringLatin1.charAt(StringLatin1.java:47)
    at java.base/java.lang.String.charAt(String.java:693)
    at Main.findAnagrams(Main.java:45)
    at Main.main(Main.java:15)
```

It is failing on this line 45.
And clearly it is due to having no content in the StringBuilder.
There is no need to handle this as an exception, rather it is more sensible to create an if loop around the main code in the nested for loop.
The loop should only execute if there is content in the sb, since we are effectively deriving indexes based on contents.

```
for (int pos=0; pos<=sb.length(); pos++)
45                System.out.println("Checking character: " + sb.toString().charAt(pos) +
```

I have included the following if statement.

```
43
44              //This is included since it will attempt to perform charAt(pos) on an empty StringBuilder
45              if (!sb.toString().isEmpty())
46              {
47              System.out.println("value of pos: " + pos);
48              System.out.println("value of sb: " + sb.toString());
49              System.out.println("Checking character: " + sb.toString().charAt(pos) +
50              "  against the main String index: " + i + "("+s.charAt(i)+")"
51              + " TO index: " + (i+p.length()) +  "("+s.charAt((i + (p.length()-1)))+")" );
52
```

And finally, it seems like the code wants to exit the inner while loop...
However I will need to examine it closely to understand if there is any excessive executions....
But it is running a tremendous of executions as below...

I am below taking a small section, but it is enough to ensure that I can understand and remediate

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: cbaebabacd //This is ok

String (p) template word: abc //This is ok

value of pos: 0 //This is ok. This refers to pos which will be index of the StringBuilder (this variable will always remain as 0)..

value of sb: abc //This is ok (it shows that StringBuilder contains String p)

Checking character: a against the main String index: 0(c) TO index: 3(a)
//it can be seen that a correction is required with output since it should mention index: 2

char found: a //this is fine, it would find the a at index 2 on the main String... Perhaps it would also require something indicating this. However it might not serve correct purpose since there

This is current StringBuilder (String p): bc //we can see it has clearly removed the a.. Perhaps there should be a screen output prior to this

value of pos: 0 //as explained above, this value will not change in any of the executions

value of sb: bc

Checking character: b against the main String index: 0(c) TO index: 3(a) //since the String is immutable, we are not able to modify it and always continue searching at same width as p.length(). Once again the error of the index is fixed as per above

char found: b

This is current StringBuilder (String p): c //This is the remaining character

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 0(c) TO index: 3(a)  //similar concept to above

char found: c

This is current StringBuilder (String p):  //We can see that StringBuilder is currently empty

REACH1  //We can see it seems to complete another execution. And since there is repetition of REACH1, it means that code is still within here:
for (int i=startPos; i<s.length();i++)

value of i: 0

value of startPos: 3  //we can also see that startPos has reached 3....  But we know that for the next anagram check, the StringBuilder has to be set to default. And we would increment the position in main String by 1. So clearly adjustment is needed

REACH1

value of i: 1

value of startPos: 3

REACH1

value of i: 2

value of startPos: 3

REACH1

value of i: 3

value of startPos: 3

REACH1

value of i: 4

value of startPos: 3

REACH1

value of i: 5

value of startPos: 3

REACH1

value of i: 6

value of startPos: 3

REACH1

value of i: 7

value of startPos: 3

REACH1

value of i: 8

value of startPos: 3

REACH1

value of i: 9

value of startPos: 3

REACH2

is an anagram of: abc //we expected the bit before the is to have the substring from the main String, this is something that requires adjusting

value of pos: 0

value of sb: abc //it has correctly restored the StringBuilder

Checking character: a against the main String index: 3(e) TO index: 6(a) //as mentioned before, the index 3 is incorrect. It still refers to old startPos.. I will need to create a variable counter to exclusively ascertain when it has completed a full cycle (i.e when it has reset the StringBuilder back to its default). This will ensure that in the main for loop, int i=0 is related to counter

NO MATCH FOUND //we can also see this is incorrect since there clearly is a match //And from hereon the code continues to loop endlessly as below.....

REACH1

value of i: 3

value of startPos: 4

REACH1

value of i: 4

value of startPos: 4

REACH1

value of i: 5

value of startPos: 4

REACH1

value of i: 6

value of startPos: 4

REACH1

value of i: 7

value of startPos: 4

REACH1

value of i: 8

value of startPos: 4

REACH1

value of i: 9

value of startPos: 4

REACH2

REACH1

value of i: 4

value of startPos: 4

REACH1

value of i: 5

value of startPos: 4

REACH1

value of i: 6

value of startPos: 4

REACH1

value of i: 7

value of startPos: 4

REACH1

value of i: 8

value of startPos: 4

REACH1

value of i: 9

value of startPos: 4

REACH2

REACH1

So taking all this in mind, there are several changes required..
The most easiest are on the screen ouputs...

However below I am now placing some of the major changes in the code

# TEST CASE: Implementing code changes to resolve issues described above

```
findAnagrams("cbaebabacd", "abc");
```

```
32      do
33      {
34          //this is going through the main String...
35          //it is starting more inwards once it has processed the entire length of p
36          //or if it fails to find a letter in String p within the main string
37          //counter has been incremented in this circumstance outside of the inner do while loop
38          //it has to be remembered the outer do while loop is when there are insufficient characters in main String
39          //to compensate for String p
40
41          startPos=counter;
42
43          for (int i=startPos; i<s.length();i++)
44          {
```

```
82                  else
83                  {
84                      System.out.println("NO MATCH FOUND");
85                      pos=p.length();
86
87                      //I have removed this since we can perform this operation when the inner do while loop exits
88                      //(when the StringBuilder is empty)
89                      //startPos++;
90
91                      System.out.println("StringBuilder being emptied: " + sb.toString());
92                      sb.delete(0,sb.length());
93                      hasCharFound=false;
94                      break;
```

```
112
113         if (sb.toString().isEmpty() && hasCharFound)
114         {
115             System.out.println(s.substring(startPos,p.length()) + " is an anagram of: " + p);
116         }
117
118         //we need to restore the original StringBuilder to check further in the String s
119         sb=sbBackup;
120
121         hasCharFound=false;
```

```
118         //we need to restore the original StringBuilder to check further in the String s
119         sb=sbBackup;
120
121         hasCharFound=false;
122
123        //we also need to make the startPos to be i+1
124         //since we are now ready to start anagram check at next position in main String
125         //since i has been affected in past execution in relation to startPos, there should be a counter variable running
126         //inside the main for loop... This variable would be incremented.
127
128         counter++;
129
130         //this while statement is so that if there are insufficient characters left in s
131         //in relation to length of p, there is no point of performing anagram check
132         //it is extremely tricky to know if this statement is 100% index perfect.
133         //but all are referencing non-zero indexed based values..
134         //startPos is no longer zero indexed based when its context is taken out of the indexing above.
135
136     }while(p.length()+startPos<s.length());
```

```
113         if (sb.toString().isEmpty() && hasCharFound)
114         {
115             System.out.println(s.substring(counter,(counter+p.length())) + " is an anagram of: " + p);
116         }
```

# TEST EXECUTION: LATEST IMPLEMENTATIONS AS ABOVE

```
findAnagrams("cbaebabacd", "abc");
```

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: cbaebabacd

String (p) template word: abc

This is STARTPOS: 0

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 0(c) TO index: 2(a)  //this has now been fixed

char found: a    at index: 2

a has been removed from StringBuilder (String p)  //now getting correct messages about removing character from StringBuilder

This is current StringBuilder (String p): bc  //This is ok

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 0(c) TO index: 2(a)  //This is ok

char found: b  at index: 1

b has been removed from StringBuilder (String p)  //This is ok

This is current StringBuilder (String p): c  //This is ok

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 0(c) TO index: 2(a)  //This is ok

char found: c  at index: 0

c has been removed from StringBuilder (String p)

This is current StringBuilder (String p):  //This is ok, empty StringBuilder

REACH1  //It is performing this since the sb isempty... There is lots of execution in i iteration. I do not think I can do anything since do while loop is a post check. It will only stop the action once it reaches it. These debugging messages will be switched off eventually.

value of i: 0

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 1

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 2

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 3

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 4

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 5

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 6

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 7

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 8

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 9

value of startPos: 3

REACH2: 3

cba is an anagram of: abc **//now resolved**

This is STARTPOS: 1 //this has now increased successfully

value of pos: 0

value of sb: abc //this has now populated successfully

Checking character: a against the main String index: 1(b) TO index: 3(e) //bae, these are the correct indexes

char found: b at index: 1 // this is correct, need to remember the indexes are still in relation to the main String s

a has been removed from StringBuilder (String p) //this is ok

This is current StringBuilder (String p): bc //this is ok

value of pos: 0 //this is ok

value of sb: bc //this is ok

Checking character: b  against the main String index: 1(b) TO index: 3(e)  //this is ok

char found: c    at index: 0  //this is incorrect.  We know the initial main String s was cbaebabacd. The fact that it is finding a match at index 0 means that it is performing indexOf on a substring which is not starting from correct index. This requires immediate remediation..

```
System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt(pos))) + "    at index: "
+ s.substring(startPos,p.length()).indexOf(p.charAt(pos)));
```

OVERCOMPLICATED

```
//if there is a match
System.out.println("*******: " + p.charAt(pos));
if (s.substring(startPos,p.length()).indexOf(p.charAt(pos))!=-1)
{
    //System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt(pos))));

    System.out.println("char found: " + p.charAt(pos) + "    at index: "
    + s.substring(startPos,p.length()).indexOf(p.charAt(pos)));
```

SIMPLIFIED

DUE TO CRITICAL NATURE OF OUTPUT, I HAVE CREATED A FRESH OUTPUT AS BELOW.

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: cbaebabacd

String (p) template word: abc

This is STARTPOS: 0

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 0(c) TO index: 2(a)

char found: a    at index: 2

a has been removed from StringBuilder (String p)

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 0(c) TO index: 2(a)

char found: a    at index: 1

b has been removed from StringBuilder (String p)

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 0(c) TO index: 2(a)

char found: a    at index: 0

c has been removed from StringBuilder (String p)

This is current StringBuilder (String p):

REACH1

value of i: 0

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 1

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 2

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 3

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 4

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 5

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 6

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 7

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 8

value of startPos: 3

This is STARTPOS: 3

REACH1

value of i: 9

value of startPos: 3

REACH2: 3

cba is an anagram of: abc

This is STARTPOS: 1 //ALL ABOVE TAKEN TO BE CORRECT FROM PREVIOUS TEST CASE

value of pos: 0

value of sb: abc

Checking character: a against the main String index: 1(b) TO index: 3(e) //substring is bae

char found: a at index: 1

a has been removed from StringBuilder (String p)

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b against the main String index: 1(b) TO index: 3(e)

char found: a at index: 0 //this still appears to be a false screen output
I have re-evaluated the logic closely and this time found the issue...
I seemed to be mixing the indexes with String p. However the inner for loop was

designed to use pos with the StringBuilder. It is resolved now!

```
if (s.substring(startPos,p.length()).indexOf(sb.toString().charAt(pos))!=-1)
{
    //System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.charAt(pos))));

    System.out.println("char found: " + sb.toString().charAt(pos) + "    at index: "
    + s.substring(startPos,p.length()).indexOf(p.charAt(pos)));

    //it removes character from the StringBuilder(which holds String p)
    //at index which it found the match

    System.out.println(sb.toString().charAt(pos) + " has been removed from StringBuilder (String p)");
    sb.deleteCharAt(pos);
    startPos++;
    System.out.println("This is current StringBuilder (String p): " + sb.toString());
    hasCharFound=true;
}
```

b has been removed from StringBuilder (String p)

## TEST CASE:  Now running code again

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: cbaebabacd

String (p) template word: abc

This is STARTPOS: 0

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: a    at index: 2

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: b    at index: 2

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: c    at index: 2

c has been removed from StringBuilder (String p)= c

This is current StringBuilder (String p):

REACH1

value of i: 0

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 1

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 2

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 3

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 4

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 5

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 6

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 7

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 8

value of startPos: 0

This is STARTPOS: 0

REACH1

value of i: 9

value of startPos: 0

REACH2: 0

cba is an anagram of: abc  Index(0)  //THIS IS OK

This is STARTPOS: 1

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 1(b) TO index: 3(e)  //I WILL BE CHECKING OUTCOME FROM HEREON

SUBSTRING EXAMINED: bae

char found: a  at index: 1

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 1(b) TO index: 3(e)

SUBSTRING EXAMINED: bae

char found: b  at index: 1

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 1(b) TO index: 3(e)

SUBSTRING EXAMINED: bae

NO MATCH FOUND  //THIS IS OK, the substring does not contain ‘c’

StringBuilder being emptied: c

REACH1

value of i: 1

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 2

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 3

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 4

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 5

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 6

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 7

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 8

value of startPos: 1

This is STARTPOS: 1

REACH1

value of i: 9

value of startPos: 1

REACH2: 1 //AT THIS POINT IT HAS EXITED THE INNER DO WHILE LOOP AND IT HAS INCREASED THE VALUE OF COUNTER SINCE IT WILL BE STARTING ANAGRAM CHECK AT INCREMENTED VALUE OF THE MAIN STRING. IT ALSO RESTORES ORIGINAL STRINGBUILDER

*****RESTORING BACKUP OF STRINGBUILDER (String p)

This is STARTPOS: 2 //IT IS JUST UNKNOWN WHY IT SKIPS THE ENTIRE INNER FOR LOOP

REACH1

value of i: 2

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 3

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 4

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 5

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 6

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 7

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 8

value of startPos: 2

This is STARTPOS: 2

REACH1

value of i: 9

value of startPos: 2

REACH2: 2

*****RESTORING BACKUP OF STRINGBUILDER (String p)

This is STARTPOS: 3 //IT IS EXACTLY SAME CASE RIGHT UP TO STARTPOS 7. BUT WE KNOW THAT IT HAS AT LEAST PERFORMED CORRECT OPERATION AND FINISHED SINCE WE KNOW THAT length of String p is 3 characters.
If it commenced at StartPos=8, it would finish StartPos=10. This would exceed limit of the index for main String with 10 characters

*****RESTORING BACKUP OF STRINGBUILDER (String p)

This is STARTPOS: 7

REACH1

value of i: 7

value of startPos: 7

This is STARTPOS: 7

REACH1

value of i: 8

value of startPos: 7

This is STARTPOS: 7

REACH1

value of i: 9

value of startPos: 7

REACH2: 7

*****RESTORING BACKUP OF STRINGBUILDER (String p)

** Process exited - Return Code: 0 **

I found the following technique was not giving the desired effect at all and still resulted in StringBuilder being empty:

```
System.out.println("*****RESTORING BACKUP OF STRINGBUILDER (String p): " + sb.toString());
sb=sbBackup;
```

My understanding that since both are object references, performing = would be storing the location in memory to which sbBackup is referencing.. And we know that in that address, it had an instance of the StringBuilder which contains String p

But it failed to function.

It later became obvious.. Since I was in effect adjusting sb to be same memory location as sbBackup... So once I had assigned it, I would effectively be sharing same location and changing contents in the backup location also.

So I went back to my known usual technique for re-instantiation.

```
System.out.println("*****RESTORING BACKUP OF STRINGBUILDER (String p): " + sb.toString());
sb=new StringBuilder(p);
```

I also observed a fairly fundamental error
I found the whole exercise to had extensive concatenation, so it was always possible these errors would occur.

```
63
64          //if there is a match
65
66          System.out.println("SUBSTRING EXAMINED: " + s.substring(startPos,(startPos+p.length())));
67
68
69          if (s.substring(startPos,(startPos+p.length())).indexOf(sb.toString().charAt(pos))!=-1)
70          {
71              //System.out.println("char found: " + s.charAt(s.substring(startPos,p.length()).indexOf(p.ch
72
73              System.out.println("char found: " + sb.toString().charAt(pos) + "    at index: "
74              + s.substring(startPos,(startPos+p.length())).indexOf(sb.toString().charAt(pos)));
75
76
77
Ln: 64, Col: 42
```

I ALSO DISCOVERED THAT I WAS MAKING A FUNDEMENTAL MISTAKE AND TOTALLY MISSING THE FOLLOWING VARIABLE

# TEST CASE: Executing entire code again

```
findAnagrams("cbaebabacd", "abc");
```

I have now unified all my remediations, better screen outputs.
Once I am certain the code is behaving, I can aim to switch off excessive messages

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: cbaebabacd

String (p) template word: abc

This is STARTPOS: 0

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: a    at index: 2

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: b    at index: 1

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: c    at index: 0

c has been removed from StringBuilder (String p)= c

This is current StringBuilder (String p):

REACH1 //I now need to find a technique to ensure these are not outputted.. It can be seen that sb is empty.... So I can add relevant condition

I have incorporated the following:

```
42
43          for (int i=startPos; i<s.length();i++)
44          {
45               if(!sb.toString().isEmpty())
46              {
47              System.out.println("This is STARTPOS: " + startPos);
48              System.out.println("This is Sb length: " + sb.length());
49              }
50
51              for (int pos=0; pos<=sb.length(); pos++)
```

```
110             if(!sb.toString().isEmpty())
111             {
112             System.out.println("REACH1");
113             System.out.println("value of i: " + i);
114             System.out.println("value of startPos: " + startPos);
115             }
```

value of i: 0

value of startPos: 0

This is STARTPOS: 0

This is Sb length: 0

REACH1

value of i: 1

value of startPos: 0

This is STARTPOS: 0

This is Sb length: 0

The code is now much more refined.. All my previously thought process of issue was in relation to do while loop was completely wrong.

## TEST CASE: Executing entire code again removing outputs no relating to meaningful logic undertaken

```
findAnagrams("cbaebabacd", "abc");
```

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: cbaebabacd

String (p) template word: abc

This is STARTPOS: 0 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: a    at index: 2

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: b    at index: 1

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 0(c) TO index: 2(a)

SUBSTRING EXAMINED: cba

char found: c    at index: 0

c has been removed from StringBuilder (String p)= c

This is current StringBuilder (String p):

REACH2: 0

cba is an anagram of: abc  Index(0) //Meaningful information of anagram

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 1 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 1(b) TO index: 3(e)

SUBSTRING EXAMINED: bae

char found: a  at index: 1

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 1(b) TO index: 3(e)

SUBSTRING EXAMINED: bae

char found: b  at index: 0

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 1(b) TO index: 3(e)

SUBSTRING EXAMINED: bae

NO MATCH FOUND  //informing end user no match

StringBuilder being emptied: c //it is emptying StringBuilder which ensures it can break out of the inner do while loop

REACH2: 1  //indication it has left the inner do while loop

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 2

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 2(a) TO index: 4(b)

SUBSTRING EXAMINED: aeb

char found: a  at index: 0

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 2(a) TO index: 4(b)

SUBSTRING EXAMINED: aeb

char found: b  at index: 2

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 2(a) TO index: 4(b)

SUBSTRING EXAMINED: aeb

NO MATCH FOUND

StringBuilder being emptied: c

REACH2: 2

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 3 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 3(e) TO index: 5(a)

SUBSTRING EXAMINED: eba

char found: a  at index: 2

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 3(e) TO index: 5(a)

SUBSTRING EXAMINED: eba

char found: b  at index: 1

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 3(e) TO index: 5(a)

SUBSTRING EXAMINED: eba

NO MATCH FOUND

StringBuilder being emptied: c

REACH2: 3

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 4 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 4(b) TO index: 6(b)

SUBSTRING EXAMINED: bab

char found: a    at index: 1

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 4(b) TO index: 6(b)

SUBSTRING EXAMINED: bab

char found: b    at index: 0

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 4(b) TO index: 6(b)

SUBSTRING EXAMINED: bab

NO MATCH FOUND

StringBuilder being emptied: c

REACH2: 4

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 5 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 5(a) TO index: 7(a)

SUBSTRING EXAMINED: aba

char found: a    at index: 0

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 5(a) TO index: 7(a)

SUBSTRING EXAMINED: aba

char found: b    at index: 1

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 5(a) TO index: 7(a)

SUBSTRING EXAMINED: aba

NO MATCH FOUND

StringBuilder being emptied: c

REACH2: 5

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 6 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 6(b) TO index: 8(c)

SUBSTRING EXAMINED: bac

char found: a    at index: 1

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 6(b) TO index: 8(c)

SUBSTRING EXAMINED: bac

char found: b    at index: 0

b has been removed from StringBuilder (String p)= bc

This is current StringBuilder (String p): c

value of pos: 0

value of sb: c

Checking character: c  against the main String index: 6(b) TO index: 8(c)

SUBSTRING EXAMINED: bac

char found: c    at index: 2

c has been removed from StringBuilder (String p)= c

This is current StringBuilder (String p):

REACH2: 6

bac is an anagram of: abc  Index(6)

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

This is STARTPOS: 7 //There is now only one occurrence of this, so its good news

This is Sb length: 3

value of pos: 0

value of sb: abc

Checking character: a  against the main String index: 7(a) TO index: 9(d)

SUBSTRING EXAMINED: acd

char found: a    at index: 0

a has been removed from StringBuilder (String p)= abc

This is current StringBuilder (String p): bc

value of pos: 0

value of sb: bc

Checking character: b  against the main String index: 7(a) TO index: 9(d)

SUBSTRING EXAMINED: acd

NO MATCH FOUND

StringBuilder being emptied: bc

REACH2: 7

*****RESTORING BACKUP OF STRINGBUILDER (String p): abc

AND AS EXPLAINED BEFORE, GIVEN THAT MAIN STRING HAS 10 CHARACTERS (index 0-9) and String p is length =3, execution has correctly terminated at startPos=7

** Process exited - Return Code: 0 **

I will now integrate all the test cases as per the challenge

## TEST CASE: I will include less commentary

```
findAnagrams("abab", "ab");
```

```
findAnagrams("abab", "ab") → [0, 1, 2]
// Anagrams: "ab", "ba", "ab"
```

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: abab

String (p) template word: ab

This is STARTPOS: 0

This is Sb length: 2

value of pos: 0

value of sb: ab

Checking character: a against the main String index: 0(a) TO index: 1(b)

SUBSTRING EXAMINED: ab

char found: a at index: 0

a has been removed from StringBuilder (String p)= ab

This is current StringBuilder (String p): b

value of pos: 0

value of sb: b

Checking character: b against the main String index: 0(a) TO index: 1(b)

SUBSTRING EXAMINED: ab

char found: b  at index: 1

b has been removed from StringBuilder (String p)= b

This is current StringBuilder (String p):

REACH2: 0

ab is an anagram of: ab  Index(0)  //This is correct

*****RESTORING BACKUP OF STRINGBUILDER (String p): ab

This is STARTPOS: 1

This is Sb length: 2

value of pos: 0

value of sb: ab

Checking character: a  against the main String index: 1(b) TO index: 2(a)

SUBSTRING EXAMINED: ba

char found: a  at index: 1

a has been removed from StringBuilder (String p)= ab

This is current StringBuilder (String p): b

value of pos: 0

value of sb: b

Checking character: b  against the main String index: 1(b) TO index: 2(a)

SUBSTRING EXAMINED: ba

char found: b  at index: 0

b has been removed from StringBuilder (String p)= b

This is current StringBuilder (String p):

REACH2: 1

ba is an anagram of: ab  Index(1) //This is correct

*****RESTORING BACKUP OF STRINGBUILDER (String p): ab

This is STARTPOS: 2

This is Sb length: 2

value of pos: 0

value of sb: ab

Checking character: a  against the main String index: 2(a) TO index: 3(b)

SUBSTRING EXAMINED: ab

char found: a  at index: 0

a has been removed from StringBuilder (String p)= ab

This is current StringBuilder (String p): b

value of pos: 0

value of sb: b

Checking character: b  against the main String index: 2(a) TO index: 3(b)

SUBSTRING EXAMINED: ab

char found: b  at index: 1

b has been removed from StringBuilder (String p)= b

This is current StringBuilder (String p):

REACH2: 2

ab is an anagram of: ab  Index(2) //This is correct

*****RESTORING BACKUP OF STRINGBUILDER (String p): ab

** Process exited - Return Code: 0 **

TEST CASE: I will  try to perform a slightly more complex scenario

```
//TEST CASE 3:
findAnagrams("azazazazazazzzzaaazazazaz", "az");
```

Welcome to Online IDE!! Happy Coding :)

String (s) to search in: azazazazazazzzzaaazazazaz

String (p) template word: az

This is STARTPOS: 0

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 0(a) TO index: 1(z)

SUBSTRING EXAMINED: az

char found: a  at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 0(a) TO index: 1(z)

SUBSTRING EXAMINED: az

char found: z  at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 0

**az is an anagram of: az  Index(0)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 1

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 1(z) TO index: 2(a)

SUBSTRING EXAMINED: za

char found: a  at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 1(z) TO index: 2(a)

SUBSTRING EXAMINED: za

char found: z  at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 1

**za is an anagram of: az  Index(1)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 2

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 2(a) TO index: 3(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 2(a) TO index: 3(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 2

**az is an anagram of: az  Index(2)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 3

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 3(z) TO index: 4(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 3(z) TO index: 4(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 3

**za is an anagram of: az  Index(3)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 4

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 4(a) TO index: 5(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 4(a) TO index: 5(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 4

**az is an anagram of: az  Index(4)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 5

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 5(z) TO index: 6(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 5(z) TO index: 6(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 5

**za is an anagram of: az  Index(5)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 6

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 6(a) TO index: 7(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 6(a) TO index: 7(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 6

**az is an anagram of: az  Index(6)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 7

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 7(z) TO index: 8(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 7(z) TO index: 8(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 7

**za is an anagram of: az  Index(7)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 8

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 8(a) TO index: 9(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 8(a) TO index: 9(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 8

**az is an anagram of: az  Index(8)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 9

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 9(z) TO index: 10(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 9(z) TO index: 10(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 9

**za is an anagram of: az  Index(9)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 10

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 10(a) TO index: 11(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 10(a) TO index: 11(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 10

**az is an anagram of: az  Index(10)**

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 11

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 11(z) TO index: 12(z)

SUBSTRING EXAMINED: zz

NO MATCH FOUND  //finding no match as expected

StringBuilder being emptied: az

REACH2: 11

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 12

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 12(z) TO index: 13(z)

SUBSTRING EXAMINED: zz

NO MATCH FOUND

StringBuilder being emptied: az

REACH2: 12

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 13

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 13(z) TO index: 14(z)

SUBSTRING EXAMINED: zz

NO MATCH FOUND

StringBuilder being emptied: az

REACH2: 13

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 14

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 14(z) TO index: 15(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 14(z) TO index: 15(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 14

za is an anagram of: az  Index(14)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 15

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 15(a) TO index: 16(a)

SUBSTRING EXAMINED: aa

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 15(a) TO index: 16(a)

SUBSTRING EXAMINED: aa

NO MATCH FOUND

StringBuilder being emptied: z

REACH2: 15

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 16

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 16(a) TO index: 17(a)

SUBSTRING EXAMINED: aa

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 16(a) TO index: 17(a)

SUBSTRING EXAMINED: aa

NO MATCH FOUND

StringBuilder being emptied: z

REACH2: 16

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 17

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 17(a) TO index: 18(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 17(a) TO index: 18(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 17

az is an anagram of: az  Index(17)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 18

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 18(z) TO index: 19(a)

SUBSTRING EXAMINED: za

char found: a  at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 18(z) TO index: 19(a)

SUBSTRING EXAMINED: za

char found: z  at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 18

za is an anagram of: az  Index(18)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 19

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 19(a) TO index: 20(z)

SUBSTRING EXAMINED: az

char found: a  at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 19(a) TO index: 20(z)

SUBSTRING EXAMINED: az

char found: z  at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 19

az is an anagram of: az  Index(19)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 20

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 20(z) TO index: 21(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 20(z) TO index: 21(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 20

za is an anagram of: az  Index(20)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 21

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 21(a) TO index: 22(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 21(a) TO index: 22(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 21

az is an anagram of: az  Index(21)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 22

This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 22(z) TO index: 23(a)

SUBSTRING EXAMINED: za

char found: a    at index: 1

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 22(z) TO index: 23(a)

SUBSTRING EXAMINED: za

char found: z    at index: 0

z has been removed from StringBuilder (String p)= z

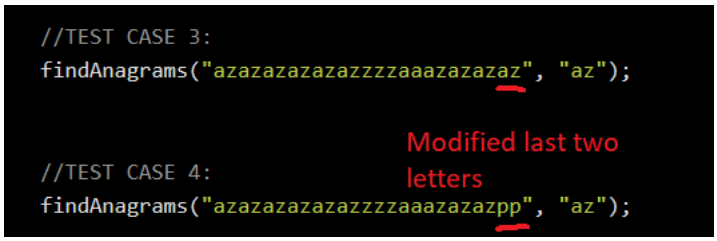
This is current StringBuilder (String p):

REACH2: 22

za is an anagram of: az  Index(22)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

This is STARTPOS: 23  //This is the last entry.... And just to be sure that it is correctly finishing at right location, I have created another test case as below...




This is Sb length: 2

value of pos: 0

value of sb: az

Checking character: a  against the main String index: 23(a) TO index: 24(z)

SUBSTRING EXAMINED: az

char found: a    at index: 0

a has been removed from StringBuilder (String p)= az

This is current StringBuilder (String p): z

value of pos: 0

value of sb: z

Checking character: z  against the main String index: 23(a) TO index: 24(z)

SUBSTRING EXAMINED: az

char found: z    at index: 1

z has been removed from StringBuilder (String p)= z

This is current StringBuilder (String p):

REACH2: 23

az is an anagram of: az  Index(23)

*****RESTORING BACKUP OF STRINGBUILDER (String p): az

** Process exited - Return Code: 0 **

# TEST CASE 4:  Based on test case 3, but changed last two letters to ensure code is running as expected

```
//TEST CASE 3:
findAnagrams("azazazazazazzzzaaazazazaz", "az");


//TEST CASE 4:
findAnagrams("azazazazazazzzzaaazazazpp", "az");
```

Modified last two letters

```
This is STARTPOS: 23
This is Sb length: 2
value of pos: 0
value of sb: az
Checking character: a  against the main String index: 23(p) TO index: 24(p)
SUBSTRING EXAMINED: pp
NO MATCH FOUND
StringBuilder being emptied: az
REACH2: 23
*****RESTORING BACKUP OF STRINGBUILDER (String p): az


** Process exited - Return Code: 0 **
```

NO ISSUES