Remove all occur in a sub-string

z/p => Star = " abbcbadd cbabbbaae "
       x x

Part = "cba"   ↓↓

o/p => Star = a bb dd bbb aae

, find ()   => to find substring in string

· erase () =>

{ while (substring exist in string)

       remove

why loop?

for Ex

abc cbebadd cba caee
       x x

abc be dd aaee
       x

a bdd aae

       final o/p

How to know   we find a string?

s.find(part) != string::npos

== npos   → not found
!= npos   → found

while (s.find (part) != string::npos)
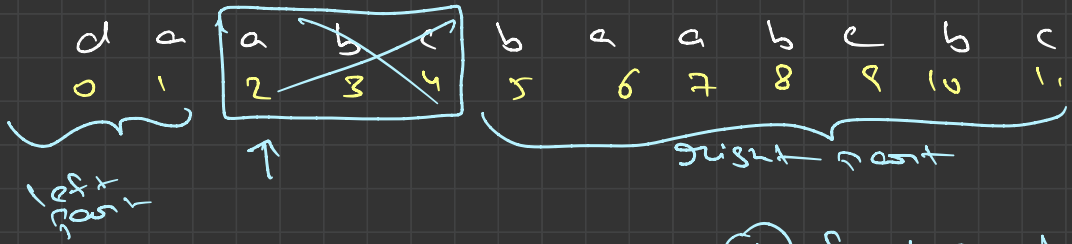 {
        s.erase (s.find (part) , part.length ());

 }
 return s;
                     ↓   ↓
               find the starting   how many
                   index   length has to
                                               be erased

Pattern = abc

d	a	a	b	c	b	a	a	b	c	b	c
0	1	2	3	4	5	6	7	8	9	10	11

left
part

↑

right part

① find part

int found = S.find (part)

more intelligent
auto size
piece

LP = S. substr (0, found)

RP = S. substr (found + part.size() , s.size())

⇒ newString = left part (concat) right part

d	a	b	a	a	b	c	b	c
0	1	2	3	4	5	6	7	8

d	a	b	a	b	c
0	1	2	3	4	5
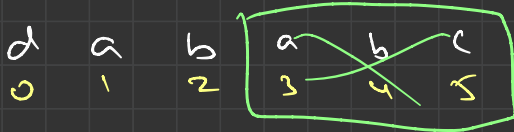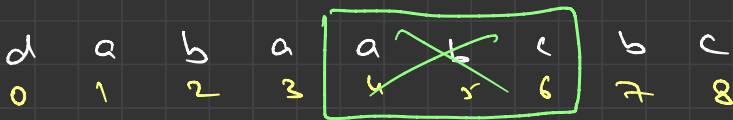
d	a	b
0	1	2

No part left
Then return
the string
as answer

```cpp
class Solution {
public:
    string removeOccurrences(string s, string part) {
        while(true){
            // found the first index of part in string s
            int found = s.find(part);

            // if not found any part break then break loop
            if(found == string::npos) break;

            // remove part from the string s

            // make a string left part from the part found
            string lp = s.substr(0, found);

            // make a string right part from the part found
            // found + part.size() -> starting index of part + part length
            string rp = s.substr(found + part.size(), s.size());

            // concat both parts
            s = lp + rp;
        }
        return s;
    }
};
```