

REPEATING STRING MATCH

★ In this problem, we are given two strings a and b , and our job is to return the number of times a needs to repeat for b to be its substring.

Obviously, b cannot be a substring of a unless the length of a repeated q times is greater than b 's length. So we ramp up a until it repeats enough q times such that its length is greater than b 's length. Then we check if it's a substring, if it isn't then we check for $q+1$ times as well.

C++ :

```
int repeatedStringMatch(string a, string b) {  
    int q = 1;  
    string s = a;  
    for (; s.length() < b.length(); q++) {  
        s += a;  
    }  
    if (s.find(b) != string::npos) {  
        return q;  
    }  
    s += a;  
}
```

```
if (s.find(w) != string::npos) {  
    return q + 1;  
}  
return -1;  
}
```