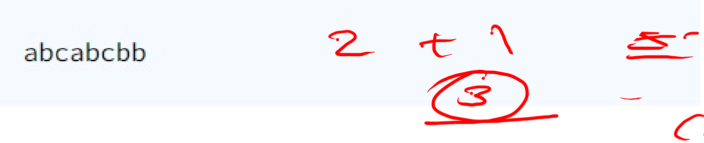
5 - 3 + 1

right - left + 1



abcabcbb

2 + 1 5

3

3 2 - 0 + 1 = 3

ch = c

| 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|
| a | b | c | a | b | c | b | b |

int left = 0; ~~1~~ ~~2~~ left 3

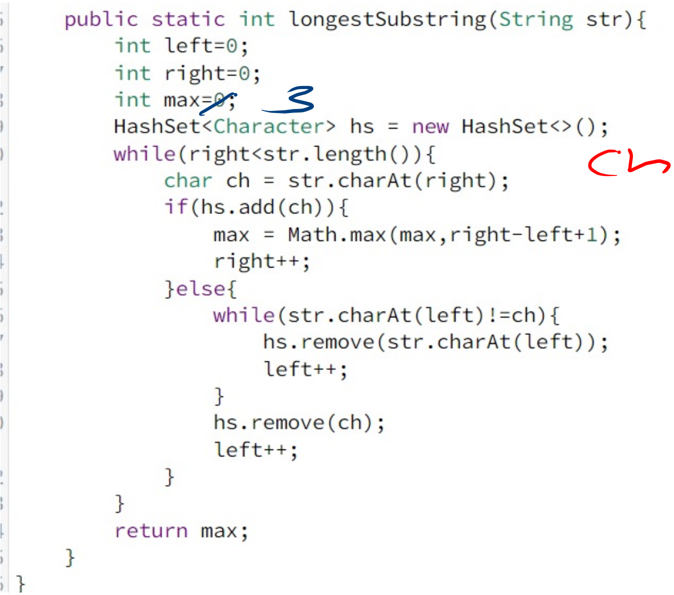
int right = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

max = ~~0~~ ~~1~~ ~~2~~ 3

"abc"

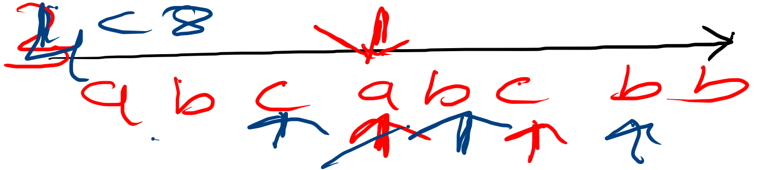
b → false

Set < ~~a~~ ~~b~~ ~~c~~ a b c



```
    public static int longestSubstring(String str){
        int left=0;
        int right=0;
        int max=0;  3
        HashSet<Character> hs = new HashSet<>();
        while(right<str.length()){
            char ch = str.charAt(right);
            if(hs.add(ch)){
                max = Math.max(max,right-left+1);
                right++;
            }else{
                while(str.charAt(left)!=ch){
                    hs.remove(str.charAt(left));
                    left++;
                }
                hs.remove(ch);
                left++;
            }
        }
        return max;
    }
}
```

ch = c



left = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

right = ~~0~~ ~~1~~ ~~2~~ ~~3~~ ~~4~~ ~~5~~ 6

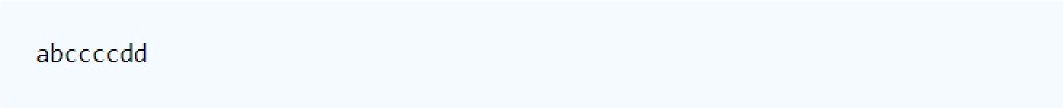
< ~~a~~, ~~b~~, ~~c~~, ~~a~~, ~~b~~, c, b

# Longest Substring Without Repeating Characters 6

```java
public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be named
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        System.out.println(longestSubstring(str));
    }
    public static int longestSubstring(String str){
        int left=0;
        int right=0;
        int max=0;
        HashSet<Character> hs = new HashSet<>();
        while(right<str.length()){
            char ch = str.charAt(right);
            if(hs.add(ch)){
                max = Math.max(max,right-left+1);
                right++;
            }else{
                while(str.charAt(left)!=ch){
                    hs.remove(str.charAt(left));
                    left++;
                }
                hs.remove(ch);
                left++;
            }
        }
        return max;
    }
}
```

c c    c c



abccccdd

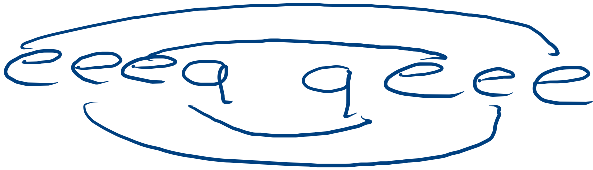
Abccccdd    a → 1

ccdAdcc ✓    b → 1    e = 6

ccabdc c ✓    q = 2

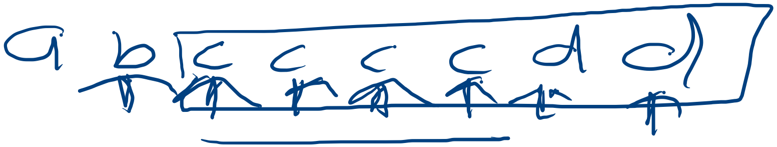
c → 4

d → 2



c c    c c

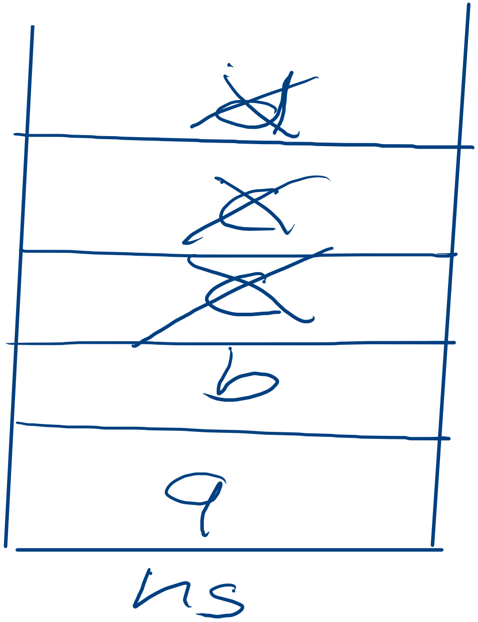
even =



len = ~~0~~ ~~2~~ ~~4~~ 6

c c d q d c c c

c c d b d c c



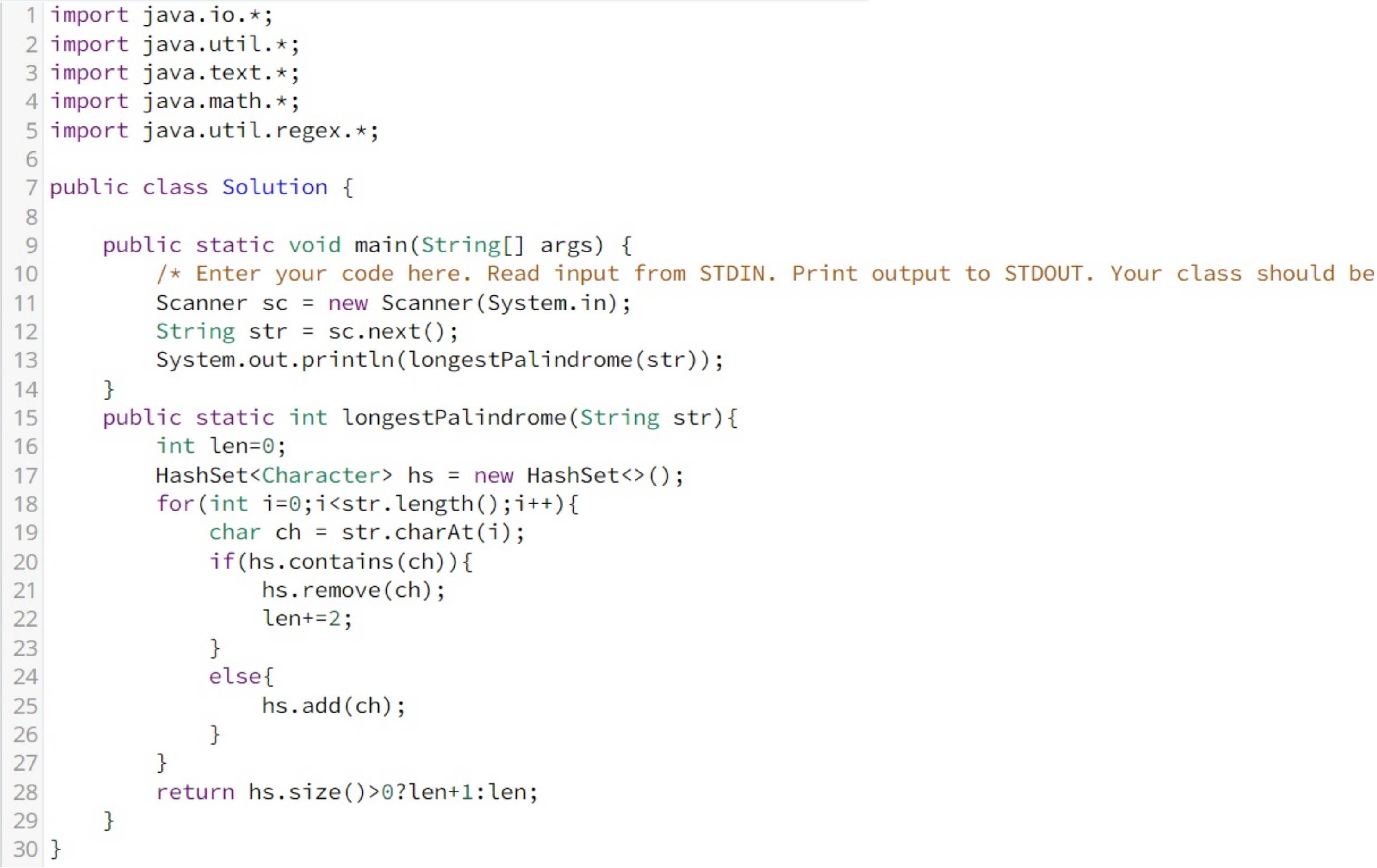
ns.size() > 0

len+1 else {

length;

}

# Length of Longest Palindrome



```
import java.io.*;
import java.util.*;
import java.text.*;
import java.math.*;
import java.util.regex.*;

public class Solution {

    public static void main(String[] args) {
        /* Enter your code here. Read input from STDIN. Print output to STDOUT. Your class should be
        Scanner sc = new Scanner(System.in);
        String str = sc.next();
        System.out.println(longestPalindrome(str));
    }
    public static int longestPalindrome(String str){
        int len=0;
        HashSet<Character> hs = new HashSet<>();
        for(int i=0;i<str.length();i++){
            char ch = str.charAt(i);
            if(hs.contains(ch)){
                hs.remove(ch);
                len+=2;
            }
            else{
                hs.add(ch);
            }
        }
        return hs.size()>0?len+1:len;
    }
}
```