 2. Given a string, return the length of the largest "block" in the string. A block is a run of adjacent chars that are the same.

maxBlock("hoopla") → 2

maxBlock("abbCCCddBBBxx") → 3

maxBlock("") → 0

Program

**import** java.util.Scanner;

**class** Triple

{

**public** **static** **int** counttriple(String str)

{

**int** count=0;

/\* int three=0;

for(int i=0;i<str.length()-1;i++)

{

if(i==0)

count++;

else

{

for(int j=i+1;j>=i;j--)

{

if(str.charAt(j)==str.charAt(j-1))

count++;

}

if(count==3)

{

three++;

count=1;

}

else

count=1;

}

}

return three;

\*/

**for**(**int** i=0;i<str.length()-2;i++)

{

**if**(str.charAt(i)==str.charAt(i+1) && str.charAt(i)==str.charAt(i+2))

count++;

}

**return** count;

}

}

**public** **class** Count\_triple {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

Scanner sc=**new** Scanner(System.***in***);

String str=sc.next();

Triple t=**new** Triple();

System.***out***.println(t.*counttriple*(str));

}

}

Output:

countTriple("abcXXXabc") → 1  
countTriple("xxxabyyyycd") → 3  
countTriple("a") → 0