

***** OUTPUT *****

TEST SCENARIO 1 AS PER EXAMPLE => PASS

```
Welcome to Online IDE!! Happy Coding :)
Original url: https://www.youtube.com/watch?v=XPEr1cArWRg
The video ID is: XPEr1cArWRg

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

TEST SCENARIO 2 AS PER EXAMPLE => PASS

```
Welcome to Online IDE!! Happy Coding :)
Original url: https://youtu.be/BCDEDi5gDPo
The video ID is: BCDEDi5gDPo

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

TEST SCENARIO 3 AS PER EXAMPLE => PASS

```
Welcome to Online IDE!! Happy Coding :)
Original url: https://youtube.com/watch?t=4m40s&v=vxP3bY-XxY4
The video ID is: vxP3bY-XxY4

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

INCORRECT URL: NO / or =

```
Welcome to Online IDE!! Happy Coding :)
Original url: https://youtube.comwatch?t4m40s&vvvxP3bY-XxY4
The video ID is: INCORRECT YOUTUBE URL

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

INCORRECT URL: NO / however since it states all links valid, not provided implementation for this.. it passes as valid in code

```
Welcome to Online IDE!! Happy Coding :)
Original url: https://www.youtube.comwatch?v=XPEr1cArWRg
The video ID is: XPEr1cArWRg
```

INCORRECT URL: Has 3 x =, however since it states all links valid, not provided implementation for this.. it passes as valid in code

```
Welcome to Online IDE!! Happy Coding :)
Original url: https://youtube.com/watch?t=4m40s&v=vxP3b=Y-XxY4
The video ID is: vxP3b=Y-XxY4

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

*** CODE **

```
/*
```

```
Online Java - IDE, Code Editor, Compiler
```

```
Online Java is a quick and easy tool that helps you to build, compile, test your programs
online.
```

```
*/
```

```
public class Main
```

```
{
```

```
    public static String youtubeld(String url)
```

```
    {
```

```
        int pos;
```

```
        String findSlash;
```

```
        String checkTwoEquals;
```

```
        if (url.indexOf('=')!=-1)    // if there is an equal sign
```

```
    {
```

```
        pos=url.indexOf('=');    //sets position of = sign
```

```
        //it now needs to check if there is another equals in the string....
```

```
        // If there is, string returned should be one position after second =
```

```
        checkTwoEquals = url.substring(pos+1,url.length());
```

```
        if (checkTwoEquals.indexOf('=')!=-1) // there is another equal sign
```

```
//4m40s&v=vxP3bY-XxY4
```

```
    {
```

```

        pos = checkTwoEquals.indexOf('=');

        return checkTwoEquals.substring(pos+1,checkTwoEquals.length());
// returns string from one position after = vxP3bY-XxY4

    }

    return url.substring(pos+1,url.length()); // returns string from one position after only =
// On this scenario https://www.youtube.com/watch?v=XPER1cArWRg it would
//extract XPER1cArWRg

}

else
{
// since no equals sign, the url will conform to something similar:
//https://youtu.be/BCDEDi5gDPo";
// it is not possible to extract everything after the / since this is present in https://
// A preferable start point would be searching after . onwards....

pos=url.indexOf('.');
//System.out.println(pos);

findSlash = url.substring(pos,url.length()); // new string is from . onwards. it can
//equally be shifted forwards
// by one or two positions, however the domain length is unknown....
// IT IS CERTAIN HOWEVER NO EQUAL SIGN WILL APPEAR AND ONLY ONE '/' IN THIS
PORTION OF STRING .be/BCDEDi5gDPo

if (findSlash.indexOf('/')!=-1)
{
pos=findSlash.indexOf('/'); // this will be position of / in following scenario:
//.be/BCDEDi5gDPo

return findSlash.substring(pos+1,findSlash.length()); // this will return BCDEDi5gDPo
}

}

return "INCORRECT YOUTUBE URL"; //Any other string will conform to a valid URL.
}

public static void main(String[] args) {
    System.out.println("Welcome to Online IDE!! Happy Coding :)");

    String[] urls = new String[6];

    urls[0]= "https://www.youtube.com/watch?v=XPER1cArWRg"; //correct URL
    urls[1]= "https://youtu.be/BCDEDi5gDPo"; //correct URL
    urls[2]= "https://youtube.com/watch?t=4m40s&v=vxP3bY-XxY4"; //correct URL
    urls[3]= "https://youtube.comwatch?t4m40s&vvxP3bY-XxY4"; // invalid URL no / or no
//EQUAL
    urls[4]= "https://www.youtube.comwatch?v=XPER1cArWRg"; // invalid URL no /

```

```
urls[5] = "https://youtube.com/watch?t=4m40s&v=vxP3b=Y-XxY4";; // invalid URL has  
three =
```

```
    for (String s: urls)  
    {  
        System.out.println("Original url: " + s);  
        System.out.println("The video ID is: " + youtubeId(s) + "\n" );  
    }  
}  
}
```