

# 牛客网-华为机试练习题 88

## 题目描述

密码按如下规则进行计分，并根据不同的得分为密码进行安全等级划分。

- 一、密码长度：
  - 5 分：小于等于4 个字符
  - 10 分：5 到7 字符
  - 25 分：大于等于8 个字符
- 二、字母：
  - 0 分：没有字母
  - 10 分：全都是小（大）写字母
  - 20 分：大小写混合字母
- 三、数字：
  - 0 分：没有数字
  - 10 分：1 个数字
  - 20 分：大于1 个数字
- 四、符号：
  - 0 分：没有符号
  - 10 分：1 个符号
  - 25 分：大于1 个符号
- 五、奖励：
  - 2 分：字母和数字
  - 3 分：字母、数字和符号
  - 5 分：大小写字母、数字和符号
- 最后的评分标准：
  - $\geq 90$ ：非常安全
  - $\geq 80$ ：安全 (Secure)
  - $\geq 70$ ：非常强
  - $\geq 60$ ：强 (Strong)
  - $\geq 50$ ：一般 (Average)

- \>= 25: 弱 (weak)
- \>= 0: 非常弱

对应输出为:

```

VERY_WEAK,

WEAK,

AVERAGE,

STRONG,

VERY_STRONG,

SECURE,

VERY_SECURE

```

- 请根据输入的密码字符串, 进行安全评定。
- 注:
- 字母: a-z, A-Z
- 数字: 0-9
- 符号包含如下: (ASCII码表可以在UltraEdit的菜单view->ASCII Table查看)
- !"#\$%&'()\*+,-./ (ASCII码: x21~0x2F)
- :;<=>?@ (ASCII码: x3A~0x40)
- [\]^\_` (ASCII码: x5B~0x60)
- {||}~ (ASCII码: x7B~0x7E)

接口描述:

Input Param

String pPasswordStr: 密码, 以字符串方式存放。

Return Value

根据规则评定的安全等级。

```

public static SafeLevel GetPwdSecurityLevel(String pPasswordStr)
{
    /*在这里实现功能*/
    return null;
}

```

## 输入描述:

输入一个string的密码

## 输出描述:

输出密码等级

示例1

输入

38\$@NoNoNo

输出

VERY\_SECURE

## 解决代码:

```
import java.util.*;
import java.io.*;

public class Main {
    public static void main(String[] args) throws Exception {
        BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
        String line = "";
        while((line=br.readLine())!=null){
            GetPwdSecurityLevel(line);
        }
    }

    private static void GetPwdSecurityLevel(String line) {
        int lenGirad = 0;
        int numGirad = 0;
        int charGirad = 0;
        int otherCharGirad = 0;
        int rewardGirad = 0;

        char[] chs = line.toCharArray();
        if(chs.length<=4)
            lenGirad = 5;
        else if(chs.length >= 4 && chs.length <= 7)
            lenGirad = 10;
        else if(chs.length >= 8)
            lenGirad = 25;

        int Numbercount = 0;
        int LowerCcount = 0;
        int UpperCcount = 0;
        int otherCcount = 0;
        for(int i=0;i<chs.length;++i)
        {
            if(chs[i]>='a' && chs[i] <='z')
                ++LowerCcount;
            else if(chs[i]>='A' && chs[i] <='Z')
                ++UpperCcount;
            else if(chs[i]>='0' && chs[i] <='9')
                ++Numbercount;
        }
    }
}
```

```

        else
            ++otherCcount;
    }

    if(Numbercount==0)
        numGirad = 0;
    else if(Numbercount==1)
        numGirad = 10;
    else
        numGirad = 20;

    if(LowerCcount==0 && UpperCcount==0)
        charGirad = 0;
    else if((LowerCcount!=0&&UpperCcount==0) || (LowerCcount==0 && UpperCcount!=0))
        charGirad = 10;
    else
        charGirad = 20;

    if(otherCcount==0)
        otherCharGirad = 0;
    else if(otherCcount==1)
        otherCharGirad = 10;
    else
        otherCharGirad = 25;

    if(LowerCcount>0&&UpperCcount>0&&Numbercount>0&&otherCcount>0)
        rewardGirad = 5;
    else if((LowerCcount>0 || UpperCcount>0)&&Numbercount>0&&otherCcount>0)
        rewardGirad = 3;
    else if((LowerCcount>0 || UpperCcount>0)&&Numbercount>0)
        rewardGirad = 2;

    int sumGriad = lenGirad + numGirad + charGirad + otherCharGirad + rewardGirad;

    if(sumGriad>=90)
        System.out.println("VERY_SECURE");
    else if(sumGriad>=80)
        System.out.println("SECURE");
    else if(sumGriad>=70)
        System.out.println("VERY_STRONG");
    else if(sumGriad>=60)
        System.out.println("STRONG");
    else if(sumGriad>=50)
        System.out.println("AVERAGE");
    else if(sumGriad>=20)
        System.out.println("WEAK");
    else if(sumGriad>=0)
        System.out.println("VERY_WEAK");
}
}

```