

String 和 Arrays

[实验要求]:

提交形式: Email

提交时间: 2020 年 7 月 22 日零点之前

Email: homework.xjtu@outlook.com

Email 的提交格式:

主题: `{"classNO":71,"stuNO":"2170010000","name":"张一天","homeworkNO":2}`

[提交文档要求]:

文档需要有封面(封面形式按照指定的文件)

最终提交的文档转为 pdf 格式提交

除了 email 中提交的电子形式外, 还需要提交纸质版(纸质版烦请各班负责同学收齐之后交到西一楼 560)

采用 JSON 格式, 注意标点符号都是英文符号, 注意每个冒号之前双引号内的值必须要大小写与给出的内容一致。

预备知识: 通过键盘输入和文件输入

Scanner 类是 Java 一个用于接收输入数据的一个实用的类型, 该类型可以和键盘关联, 也可以和指定的文件关联, 从而达到通过键盘输入数据和通过文件输入数据的目的。

程序 **KeyboardScanner.java** 演示 Scanner 类型通过键盘输入各种基本类型的数据。该程序运行之后的输出效果如下(注意黑体字代表输入的内容):

Enter an integer: **12**

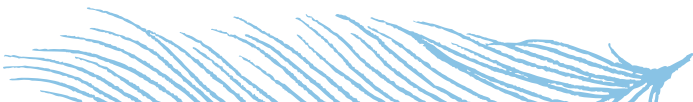
Enter a floating point number: **24.56**

Enter your name: **yuan**

Hi! yuan, the sum of 12 and 24.56 is 36.56

程序代码如下图所示。在图示中, 请自行敲代码练习 Scanner 类接收键盘输入数据的模式。并通过 Java API 的查找发现 Scanner 与输入有关的其他函数, 并加以验证练习。

另外, 在图示中, 输出函数不再使用之前经常使用的 print 或者 println 函数, 而是使用了 printf 函数, 该函数的第一个参数必定是一个字符串(用来规定输出的格式), 类似于 C 语言中 printf 函数。请大家仔细体会练习, 在以后的编码中灵活使用。



```
import java.util.Scanner; //needed to use Scanner for input
public class KeyboardScanner {

    public static void main(String[] args) {
        int num1;
        double num2;
        String name;
        double sum;

        // Setup a Scanner called in to scan the keyboard(System.in)
        Scanner in = new Scanner(System.in);
        System.out.print("Enter an integer: ");
        num1 = in.nextInt(); //use nextInt() to read int
        System.out.print("Enter a floating point number: ");
        num2 = in.nextDouble(); // use nextDouble() to read double
        System.out.print("Enter your name: ");
        name = in.next(); //use next() to read String

        //please notice:we use printf()method instead of using print() method
        //the first argument of printf() is the same as printf of C programming
        System.out.printf("Hi! %s, the sum of %d and %.2f is %.2f \n",
            name, num1, num2, (num1 + num2));
    }
}
```

程序 **FileScanner.java** 演示 Scanner 类型通过文件输入各种基本类型的数据。程序的运行输出结果如下：

Hi! yuan, the sum of 12 and 24.56 is 36.56

运行该程序需要在包含该程序的文件夹下创建一个文本文件 “in.txt”，并且保证该文件中的内容如下：

12
24.56
yuan

程序代码如下：

```
import java.util.Scanner; //Needed to use Scanner for input
import java.io.File; //Needed to use File
import java.io.FileNotFoundException; //Needed for file operation

public class FileScanner {

    public static void main(String[] args)
        throws FileNotFoundException { //Needed for file operation
        int num1;
        double num2;
        String name;
        double sum;

        // Setup a Scanner to read from a text file
        Scanner in = new Scanner( new File( "in.txt"));
        num1 = in.nextInt();
        num2 = in.nextDouble();
        name = in.next();
        System.out.printf("Hi! %s, the sum of %d and %.2f is %.2f \n",
            name, num1, num2, (num1 + num2));
        in.close(); // This operation must be done.
    }
}
```

题目 1: Run Length Encoding

编写一个使用 run-length 技术而实现的编码和解码程序。(为了使处理简化, 假设待编码的内容只包含英文字符 (区分大小写))。

Run-length encoding (RLE) 是一种简单的数据压缩 (或称为编码) 技术, runs 代表了一个连续的字符序列, 在这种编码技术下, 一个 run 就是相同字符的连续最长序列, 这个 run 被编码之后将使用 “长度+字符” 来替代原 run 串。

例如, 下面原始数据序列的长度是 53, 经过 RLE 之后则压缩到长度 13。

```
"WWWWWWWWWWWWBWWWWWWWWWWBWWWWWWWWWWBWWWWWWWWWWB" -> "12WB12W3B24WB"
```

RLE 技术可以使得原码可以无损失的从压缩码中还原:

```
"AABCCCDDEEEE" -> "2AB3CD4E" -> "AABCCCDDEEEE"
```

题目 2: 长度 n 的子序列最大乘积

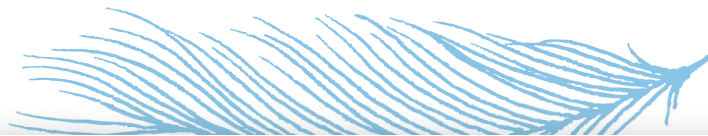
给定一个数字序列字符串, 计算给定的长度 n 的子序列中的最大乘积值。

例如: 如果输入 “1027839564”, 指定长度为 3 的最大子序列乘积值为 270 ($9*5*6$); 指定长度为 5 的最大子序列乘积值为 7560 ($7*8*3*9*5$)。

备注:

1. 数字序列字符串的最大长度 `maxLength` 的范围为: $[1..100]$;
2. n 的取值范围为 $[1..maxLength-1]$;
3. 程序要注意处理边界情况。

下图为一个长度为 100 的字符串数字序列, 在这个序列中, 长度为 4 的最大子序列乘积为 5832 ($9*9*8*9$)

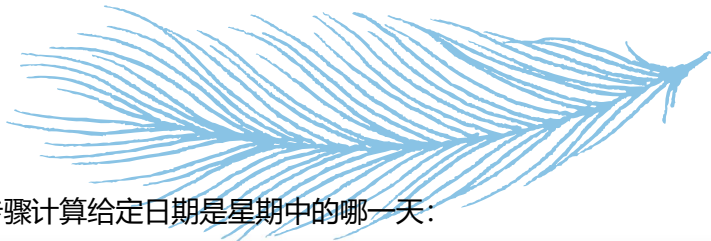


```
73167176531330624919225119674426574742355349194934
96983520312774506326239578318016984801869478851843
85861560789112949495459501737958331952853208805511
12540698747158523863050715693290963295227443043557
66896648950445244523161731856403098711121722383113
62229893423380308135336276614282806444486645238749
30358907296290491560440772390713810515859307960866
70172427121883998797908792274921901699720888093776
65727333001053367881220235421809751254540594752243
52584907711670556013604839586446706324415722155397
53697817977846174064955149290862569321978468622482
83972241375657056057490261407972968652414535100474
82166370484403199890008895243450658541227588666881
16427171479924442928230863465674813919123162824586
17866458359124566529476545682848912883142607690042
24219022671055626321111109370544217506941658960408
07198403850962455444362981230987879927244284909188
84580156166097919133875499200524063689912560717606
05886116467109405077541002256983155200055935729725
71636269561882670428252483600823257530420752963450
```

题目 3：创建一个日期工具类 DateUtil

编写一个 DateUtil 类型，该类型为与日期处理有关的工具类，该类需要完成的方法如下所列：

- `static boolean isLeapYear(int year)`: returns true if the given year is a leap year. A year is a leap year if it is divisible by 4 but not by 100, or it is divisible by 400.
- `static boolean isValidDate(int year, int month, int day)`: returns true if the given year, month and day constitute a given date. Assume that year is between 1 and 9999, month is between 1 (Jan) to 12 (Dec) and day shall be between 1 and 28|29|30|31 depending on the month and whether it is a leap year.
- `static int getDayOfWeek(int year, int month, int day)`: returns the day of the week, where 0 for SUN, 1 for MON, ..., 6 for SAT, for the given date. Assume that the date is valid.
- `static void printCalendar(int year, int month)`: The calendar for the specified year/month is output on the screen according to the calendar output format.
- `static void printCalendar(int year)`: The Calendar for the specified year is output on the screen according to the calendar output format.
- `static String formatDate(int year, int month, int day)`: prints the given date in the format "xxxdy d mmm yyyy", e.g., "Tuesday 14 Feb 2012". Assume that the given date is valid.



提示：

1.可以参看下面的步骤计算给定日期是星期中的哪一天：

1. Based on the first two digit of the year, get the number from the following "century" table.

1700-	1800-	1900-	2000-	2100-	2200-	2300-	2400-
4	2	0	6	4	2	0	6

Take note that the entries 4, 2, 0, 6 repeat.

2. Add to the last two digit of the year.

3. Add to "the last two digit of the year divide by 4, truncate the fractional part".

4. Add to the number obtained from the following month table:

	Jan	Feb	Mar	Apr	May	Jun	Jul	Aug	Sep	Oct	Nov	Dec
Non-Leap Year	0	3	3	6	1	4	6	2	5	0	3	5
Leap Year	6	2	same as above									

5. Add to the day.

6. The sum modulus 7 gives the day of the week, where 0 for SUN, 1 for MON, ..., 6 for SAT.

例如：根据上面的计算步骤，计算 2020 年 7 月 1 日属于星期几？

$(6 + 20 + 20/4 + 6 + 1) \% 7 = 3$ （星期三）


2. 日历的输出格式可以参考下面的格式：

MON	TUE	WED	THU	FRI	SAT	SUN
						1
2	3	4	5	6	7	8
9	10	11	12	13	14	15
16	17	18	19	20	21	22
23	24	25	26	27	28	29

当 DateUtil 类实现了之后，请自行编写一个程序 TestDateUtil.java 来验证每个函数执行的正确性，要求检查的尽可能全面。（可以借助 Java API 中的 Calendar 类中的函数辅助验证）

题目 4：摩斯密码

莫尔斯电码（又译为摩斯密码，Morse code）是一种时通时断的信号代码，通过不同的排列顺序来表达不同的英文字母、数字和标点符号。它发明于 1837 年，发明者有争议，是美国人塞缪尔·莫尔斯或者艾尔菲德·维尔。摩斯密码是一种早期的数字化通信形式，但是它不同于现代只使用零和一两种状态的二进制码，它的代码包括点、划、点和划之间的停顿、每个词之间的中等的停顿以及句子之间长的停顿。我们的停顿用空格字符表示，摩斯码之间的停顿为一个空格，摩斯码构成的单词和单词之间的停顿用三个空格表示。具体的摩斯编码如下：



比如“AN EGG”的摩斯码即为如下形式：

每个单词之

每个字母之间用一个空格分隔

每个单词之间用三个空格分隔

题目中提供一个包含摩斯密码的文本文件，名为：encode.txt。

请编写一个 MorseCodeDecode.java 程序，该程序的输入数据从文件 encode.txt 中读取，最终给出对该文件所包含的摩斯密码的解密结果。

扩充内容 (该项内容不作为作业要求):

编写一个 MorseCodeEncode.java 程序，该程序的输入数据为一个包含英文字符和数字的字符串，将该字符串以摩斯密码进行编码，并将编码结果输出到一个文件中。

将内容输出到文件可以使用 `PrintWriter` 类型，参考的使用语句序列如下：

```
PrintWriter pw = new PrintWriter(new File(filename));
pw.print("hello world!");
pw.close();
```