

题目描述

请在一个字符串中找到连续最长的数字串，并返回这个数字串。

如果存在长度相同的连续数字串，返回最后一个。

如果没有符合条件的字符串，返回空字符串”。

注意：

- 数字串可以由数字 0-9、小数点”.”、正负号”+”组成，长度包括组成数字串的所有符号。
- ”.”、“+”仅能出现一次，“-”的两边必须是数字，“+”仅能出现在开头且其后必须要有数字。
- 长度不定，可能含有空格。

输入描述

无

输出描述

无

用例

输入	1234567890abcde+12345 678 9ed
输出	+12345 678
说明	无

题目解析

本期只要构造出符合题目对数字串要求的正则即可。

”.”、“+”仅能出现一次，“-”的两边必须是数字，“+”仅能出现在开头且其后必须要有数字。

const regExp = /([+]?[0-9]+\.?[0-9]+)?/g;

JavaScript算法源码

```
1  /**
2  * 正则表达式 匹配数字 + 小数
3  */
4  const readline = require("readline");
5  3
6  const rl = readline.createInterface({
7  input: process.stdin,
8  output: process.stdout,
9  });
10  8
11  rl.on("line", (line) => {
12  const regExp = /([+]?[0-9]+\.?[0-9]+)?/g;
13  11
14  let ans = "";
15  13
16  while (true) {
17  let res = regExp.exec(line);
18  16
19  if (!res) break;
20  const match = res[1];
21  ans = match.length >= ans.length ? match : ans;
22  }
23  console.log(ans);
24  };
```

Java算法源码

```
1  import java.util.Scanner;
2  import java.util.regex.Matcher;
3  import java.util.regex.Pattern;
4  4
5  public class Main {
6  public static void main(String[] args) {
7  Scanner sc = new Scanner(System.in);
8  8
9  String s = sc.nextLine();
10  System.out.println(getResult(s));
11  12
12  }
13  13
14  public static String getResult(String s) {
15  Matcher matcher = Pattern.compile("(+)?[0-9]+\\.?[0-9]+").matcher(s);
16  15
17  String ans = "";
18  int maxLen = 0;
19  19
20  while (matcher.find()) {
21  String tmp = matcher.group();
22  if (tmp.length() >= maxLen) {
23  maxLen = tmp.length();
24  ans = tmp;
25  }
26  }
27  27
28  return ans;
29  }
30  }
```

Python算法源码

```
1  import re
2  2
3  # 输入字符串
4  s = input()
5  5
6  6
7  # 正则表达式
8  def getResult(s):
9  reg = re.compile("(+)?[0-9]+\\.?[0-9]+")
10  10
11  matchList = reg.findall(s)
12  matchList.sort(key=lambda x: len(x))
13  13
14  return matchList[-1]
15  15
16  # 输出结果
17  print(getResult(s))
```