

《程序设计与算法基础I》在线测试题目说明

电子科技大学 信息与软件工程学院

版本：2018-2019-1（学生版）

实验四（Experiment 4）

1. 栈（Stack）

题目编号：

10-01

题目内容：

修改10.2节的栈示例使它存储字符而不是整数。接下来，增加 `main` 函数，用来要求用户输入一串圆括号或花括号，然后指出它们之间的嵌套是否正确。

输出范例：

范例1：

```
Enter parentheses and/or braces: ()({})({})
Parentheses/braces are nested properly
```

范例2：

```
Enter parentheses and/or braces: ({}
Parentheses/braces are NOT nested properly
```

输入输出：

- 输入字符为 `(`、`)`、`{`、`}` 的组合，长度不超过 100。
- 输出若满足嵌套要求，打印 `Parentheses/braces are nested properly`，否则打印 `Parentheses/braces are NOT nested properly`；除以上两种输出二选其一，没有其他字符输出。

2. 逆序 (Reversal)

题目编号：

12-01 (a)

题目内容：

编写程序读一条消息，然后逆序打印出这条消息。

输出范例：

范例1：

```
Enter a message: Don't get mad, get even.  
Reversal is: .neve teg ,dam teg t'noD
```

范例2：

```
Enter a message: Hello, world!  
Reversal is: !dlrow ,olleH
```

输入输出：

输入为一条消息（字符串），长度 50 位以内。输出为上述消息的完全逆序。

温馨提示：

一次读取消息中的一个字符（用getchar函数），并且把这些字符存储在数组中，当数组写满或者读到字符 '\n' 时停止读入。

3. 最大最小单词简略版 (The Biggest and Smallest Words)

题目编号：

13-01

题目名称：

编写程序找出一组单词中“最小”单词和“最大”单词。用户输入单词后,程序根据字典顺序决定排在最前面和最后面的单词。当用户输入4个字母的单词时,程序停止读入。假设所有单词都不超过20个字母。

输出范例：

范例1：

```
Enter word: dog
Enter word: zebra
Enter word: rabbit
Enter word: catfish
Enter word: walrus
Enter word: cat
Enter word: fish

Smallest word: cat
Largest word zebra
```

范例2：

```
Enter word: computer
Enter word: ink
Enter word: light
Enter word:

bookrack
Enter word: book

Smallest word: book
Largest word: light
```

输入输出：

- 默认每个单词长度不超过 20 位。
- 如果没有空白符以外的输入，程序将一直等待直到获取非空白符输入。

温馨提示：

使用两个名为 `smallest_word` 和 `largest_word` 的字符串来分别记录所有输入中的“最小”单词和“最大”单词。用户每输入一个新单词,都要用 `strcmp` 函数把它与 `smallest_word` 进行比较如果新的单词比 `smallest_word` “小”,就用 `strcpy` 函数把新单词保存到 `smallest_word` 中。用类似的方式与 `largest_word` 进行比较。用 `strlen` 函数来判断用户是否输入了4个字母的单词。

评分规则：

- 编译通过20分
- 格式规范10分
- 测试用例70分

注意事项：

- C语言文件请采用utf8编码；
- 默认最后一个printf格式语句结束时都需要采用\n换行；
- 尽量避免printf语句换行（如下例）；除非格式全对，否则会被认为全错。

```
printf("a: %d\n"  
"b: %d\n"  
"c: %d\n"  
"d: %d\n",a,b,c,d);
```

////////////////////////////////////

编辑：SynX.Rumba Group

审校：张翔

创建：2018.10.02

修订：2018.11.24

////////////////////////////////////