

Challenge 2 A=B

Challenge 2 A=B

题目描述

输入格式

输出格式

关键字

指令集

示例

首尾

指令集

示例

一次性

指令集

示例

提交格式

附录：只有基础替换指令的A=B machine的运行机制

注意，本次Challenge是Assignment3中第3题A=B的后续。因此，请先完成基础的A=B替换功能，再来做Challenge 2。

本次Challenge共计10分，有10个测试案例，每个测试案例占1分。

题目描述

Assignment3中的 `A=B` 语言只包含一条最基础的替换指令。它甚至都不能打印"hello world"，怎么能被称为一门编程语言呢？

因此，本次Challenge需要同学在基础替换命令的基础上，增加更加高级的替换指令，增强 `A=B` 的计算能力，使得它能够处理更加复杂的任务。只有基础替换指令的A=B machine的运行机制请见文档最后的附录。

以下三个章节会在前面章节的基础上，增加新的指令。

- “关键字”章节引入第一个关键字 `(return)`，用于指定输出字符串并终止程序
- “首尾”章节引入关键字 `(start)` 和 `(end)`，使得程序具有处理字符串的开头和结尾的能力
- “一次性”章节引入关键字 `(once)`，支持只会执行一次的指令，提供了对字符串做预处理的能力

完成不同章节后能够通过的测试用例情况如下：

- 完成“关键字”章节后，可以通过前3个测试案例
- 完成“关键字”和“首尾”章节后，可以通过前6个测试案例
- 完成“关键字”“首尾”和“一次性”章节后，可以通过10个测试案例

输入格式

- 第1行**：待处理的输入字符串 `S`
- 第2行开始**：每行包含一条指令，每条指令的格式是 `A=B`，其中 `A` 和 `B` 是字符串

所有字符串由字母a-z,A-Z，数字0-9以及其他字符 () 组成。s 的长度 ≥ 1 ，A 和 B 可能为空串。
指令数 ≥ 1 。

输出格式

根据增加了新指令后的运行机制，按行输出程序运行时字符串 s 的变化过程。
注意，我们保证输入的测试用例一定会停机。

关键字

既然提到了打印"elloworld"的问题，我们就先来实现这个功能。
本节引入关键字 (return)，用于指定输出字符串并终止程序。

指令集

- A=B 的指令集包括：
- 1. string1=string2
尝试找到字符串最靠左的一个string1，并将其替换为string2
 - 2. string1=(return)string2
如果匹配到string1，立即终止程序运行，直接将string2作为输出串

示例

示例1: 打印elloworld

输入

```
a
=(return)elloworld
```

输出

```
a
elloworld
```

示例2: 判断每个字母的个数是否是奇数

输入

```
bccaca
ba=ab
ca=ac
cb=bc
aaa=a
bbb=b
ccc=c
aa=(return>false
bb=(return>false
cc=(return>false
=(return>true
```

输出

```
bccaca
bcacca
baccca
abccca
abccac
abcacc
abaccc
aabccc
aabc
false
```

首尾

本节引入关键字 `(start)` 和 `(end)`，使得程序具有处理字符串的开头和结尾的能力。

指令集

`A=B` 的指令集包括：

1. `string1=string2`

尝试找到字符串最靠左的一个`string1`，并将其替换为`string2`

2. `string1=(return)string2`

如果匹配到`string1`，立即终止程序运行，直接将`string2`作为输出串

3. **`(start)string1=string2`**

`(end)string1=string2`

如果字符串的头/尾部为`string1`，将其替换为`string2`

4. **`string1=(start)string2`**

`string1=(end)string2`

如果匹配到`string1`，将其移除，并在字符串头/尾部添加`string2`

示例

示例1: 去除头部和尾部的所有a

输入

```
acacabaa
(start)a=
(end)a=
```

输出

```
acacabaa
cacabaa
cacaba
cacab
```

示例2: 回文串

输入

```
bacacab
a1a1=
b1b1=
c1c1=
(start)a=(end)1a1
(start)b=(end)1b1
(start)c=(end)1c1
11=(return>false
=(return>true
```

输出

```
bacacab
acacab1b1
acaca
caca1a1
cac
ac1c1
a
1a1
true
```

一次性

本节引入关键字 `(once)`，支持只会执行一次的指令，提供对字符串做预处理的能力。

指令集

A=B 的指令集包括：

1. string1=string2

尝试找到字符串最靠左的一个string1，并将其替换为string2

2. string1=(return)string2

如果匹配到string1，立即终止程序运行，直接将string2作为输出串

3. (start)string1=string2

(end)string1=string2

如果字符串的头/尾部为string1，将其替换为string2

4. string1=(start)string2

string1=(end)string2

如果匹配到string1，将其移除，并在字符串头/尾部添加string2

5. **(once)string1=string2**

在首次成功执行（即匹配到string1）后，忽略这条命令

示例

示例1：交换第一个和最后一个字母

输入

```
abc
(once)=(start)t
(start)ta=(end)ta
(start)tb=(end)tb
(start)tc=(end)tc
at=(start)a
bt=(start)b
ct=(start)c
```

输出

```
abc
tabc
bcta
cba
```

示例2：复制输入串

输入

```
abc
(once)=(start)t
(once)=(end)x
tx=
A=(end)a
B=(end)b
C=(end)c
ta=aAt
tb=bBt
tc=cCt
```

输出

```
abc
tabc
tabcx
aAtbcx
atbcxa
abBtcxa
abtcxab
abcCtxab
abcCab
abcabc
```

提交格式

在编写完程序并测试完毕后，将文件夹1_advanced_a_b中除了main.cpp以外的文件删除之后打包压缩，最后用自己的学号命名，即可提交。

你提交的文件结构应该和以下形式**完全一样**：

```
<your student number>.zip
|- 1_advanced_a_b
|   |- main.cpp
```

附录：只有基础替换指令的A=B machine的运行机制

为保证文档的独立性，贴出只有基础替换指令的A=B machine的运行机制。这部分内容和Assignment3的A=B运行机制相同。

只有基础替换指令的A=B machine的运行机制如下：

1. 读取输入的待处理字符串 S
2. 进入执行指令阶段，准备读取第1行的指令
3. 检查是否还有未读取的指令。如果为否，跳转至7
4. 读取下一条指令 $A=B$
5. 从左往右扫描 S ，检查是否存在子串 A 。如果为否，跳转至3
6. 将 S 中的第一处子串 A 替换为 B ，并跳转至2

7. 输出当前结果 s

文字对应的流程图如下所示。

