

数字搜索

(regular)

【背景描述】

HURRICANE小组最近接到了一个搜索文本的任务，即从一个由数字构成的长文本中，匹配满足指定条件的子串。搜索的条件采用形如‘ $(0+10^*1)^*10^*$ ’这样的正则表达式来描述。其中正则表达式的归纳定义如下：

1. $0, 1, \dots, 9, 0^*, 1^*, \dots, 9^*$ 是正则表达式；
2. 如果A和B是正则表达式，则 (A) , $A+B$, AB , $(A)^*$ 都是正则表达式；
3. 只有按以上方法构造出来的表达式才是正则表达式。

其中， $A+B$ 表示“或者”关系， AB 表示“连接”关系， $(A)^*$ 表示A的内容“重复”零次或者多次。比如正则表达式 $(12+3)(4+5)6^*$ ，就可以匹配以124, 125, 34, 35之一开头，之后接零0个或任意多个6的字符串（例如字符串12566）。正则表达式 $(1+0)^*$ 可以匹配所有由0和1构成的字符串，或者是空串。如果一个正则表达式不能匹配空串，则称它是非空的。本题考虑的都是非空正则表达式。

如果在给定文本的某一个位置，存在一个以该位置结束的子串，能够被给定的非空正则表达式匹配，则称该位置是可匹配的。现在HURRICANE小组接到的任务就是找出所有可匹配的位置。你能帮助他们完成这个任务么？

【任务描述】

你的程序需要根据给定的输入，给出符合题意的输出：

- 输入包括一个满足如上定义的正则表达式，以及一长串文本；
- 你需要根据输入的正则表达式及文本，找出文本中所有可匹配的位置；
- 你给出的输出需要包括所有可匹配的位置。

【输入格式】: (regular.in)

输入文件regular.in的第一行描述一个正则表达式，第二行为需要处理的文本：

- 第一行的正则表达式包括由一个空格分开的两个部分：
 - 一个非负整数 n ($1 \leq n \leq 10$)，表示我们所要考虑的数字集（即在正则表达式和文本中所出现的数字）是 $0, 1, \dots, n-1$ 。
 - 接下来是一个正则表达式，它由 $\{‘(’, ‘)’, ‘+’, ‘*’\}$ 中的4个符号和 $\{0, \dots, n-1\}$ 中的数字构成，表达式的长度不超过500个字符。
- 第二行为一个由0到 $n-1$ 之间数字构成的字符串，为需要处理的文本。

该文本长度不超过 10,000,000 个字符。

【输入样例】

4 12333+33 312331	说明： 对于输入示例，需要处理的文本是'3123 <u>3</u> 1'，其中只有第 5 个字符所在的位置（下划线所在处）是 <u>可匹配的</u> 。这时 <u>正则表达式</u> '12333+33'中的'33'可以与之匹配。
----------------------	--

【输出格式】: (regular.out)

输出文件只有一行，包括一些由空格分开的整数，按从小到大的顺序依次输出所需处理的文本中每一个可匹配的位置。

【输出样例】

5

【运行限制】

运行时间	2 秒
内存使用	64M

【有关语言方面的提示】

对于使用 C++ 选手来说，用 ifstream 读取数据是一个很慢的过程，可能会消耗大量的时间。所以我们建议使用 fread/fgetc/fscanf 读取数据，并使用 setbuf 设置缓存。另，fscanf(file, "%c", &c) 的速度经过测试表现也比较慢，可以使用 fgetc(file) 代替。

对于使用 Pascal 的选手来说，可以用 SetTextBuf 设置缓存来加快读取文件的速度。也可以考虑使用 BlockRead 来读取文件。

【评分方法】

本题目一共有十个测试点，每个测试点的分数为总分数的 10%。对于每个测试点来说，如果你给出的答案正确，那么你将得到该测试点全部的分数，否则得 0 分。

提示：在本次的测试数据中，有 6 个测试点中的正则表达式不出现 '*'，其中有 3 个测试点，正则表达式只由数字和 '+' 构成。有一个测试点的待处理文本不超过 1,000,000 个字符。该题有严格的时间限制，请尽量优化你的算法。