

订餐

dish.c/cpp/pas

(1s/256M)

【题目描述】

这个周末，Joe 一如既往的拿出了外卖的点餐单。菜单上从上到下列着 N 个菜，每个菜 Joe 会列出一个美味值。Joe 这次希望从菜单上连续地点 K 个菜，并且美味值之和最大。

但在计算美味值之和时，Joe 有自己的一套计算方法：由于他会按照顺序享受这 K 个菜，并且越吃越后面的菜他越享受，所以对于他吃的第 i 个菜，设它的美味值为 D ，美味值之和就会加上 $D*i$ 。例如，如果他要顺序吃美味值为 3, 8, 5 的菜，那么美味值之和为 $3 + 8*2 + 5*3 = 34$ 。

Joe 给出了这 N 个菜的美味值，他想知道所能达到最大的美味值之和。

【输入格式】 dish.in

第一行两个整数 N, K ，代表菜的总数和他要选择的菜数。
第二行 N 个整数，代表从上到下菜单上菜品的美味值。

【输出格式】 dish.out

一行一个整数，代表最大的美味值之和。

【样例输入】

```
5 3
2 4 1 5 3
```

【样例输出】

```
21
```

【数据规模与范围】

对于 40% 的数据， $N, K \leq 2000$;

对于 100% 的数据， $N, K \leq 100000$ ，各菜品的美味值 D 满足 $0 \leq D \leq 1000$ 。

大奖赛

champion.c/cpp/pas

(1s/256M)

【题目描述】

Lancelot 市近期要举办大奖赛啦！住在市里的市民都十分兴奋，Morgan 也不例外。他查了一下比赛的信息，发现比赛一共由 N 场，并且每一场的门票价格可能会不相等。Morgan 留给比赛的预算是 K 元；他想知道，一共有多少种买票的方案，使得门票之和不超过约 K 呢？

【输入格式】 champion.in

第一行两个整数 N 与 K ，代表比赛的场数和自己的预算。
第二行 N 个整数 A_i ，代表每场比赛的门票价格。

【输出格式】 champion.out

一行一个整数，代表买票的总方案数。

【样例输入】

```
5 1000
2000 100 500 500 1000
```

【样例输出】

```
8
```

【数据范围与约束】

对于 30% 的数据， $N \leq 10$ 。

对于 60% 的数据， $K \leq 10,000$ 。

对于 100% 的数据， $1 \leq N \leq 40$, $0 < K, A_i \leq 1,000,000,000$ 。

秘密信息

hidden.c/cpp/pas

(1s/256M)

【题目描述】

Irene 想用以下的方法加密一条信息（这是她从密码学书上自学来的）：
假定这条信息可以用一个字符串 S 表示，其中 $S=BCAAD.$ （其中“.”代表字符串结尾）。Irene 首先把 S 的所有循环同构串写下来（所谓循环同构即是不不断地把字符串开头的字符移动到尾端）：

BCAAD.

CAAD.B

AAD.BC

AD.BCA

D.BCAA

.BCAAD

接下来她会把这些字符串都排序：

.BCAAD

AAD.BC

AD.BCA

BCAAD.

CAAD.B

D.BCAA

接下来她会把这些字符串的最后一位按照顺序写下来得到加密串 $T=DCA.BA$

现在 Irene 发现在密码学的书上写着一条来历不明的加密串 T 。她迫切地知道，如果信息是按照这种方式加密的，那么原串是什么？

【输入格式】 hidden.in

一行一个由大写字母构成，长度为 N 的字符串（但字符串有且只有一个“.”）。

【输出格式】 hidden.out

输出一行，输出原串。数据保证有唯一解。

【样例输入 1】

PH.EL

【样例输出 1】

HELP.

【样例输入 2】

BBA.AA

【样例输出 2】

ABAAB.

【数据范围与约束】

对于 20%的数据， $2 \leq N \leq 10$;

对于 50%的数据， $2 \leq N \leq 27$ ，字符串内部的字母互不相同。

对于 100%的数据， $2 \leq N \leq 30000$ 。

括号序列

brackets.c/cpp/pas

(2s/256M)

【题目描述】

课堂上，Felix 刚刚学习了关于括号序列的知识。括号序列是一个只由左括号“(”和右括号”)”构成的序列；进一步的，一个合法的括号序列是指左括号和右括号能够一一匹配的序列。

如果用规范的语言说明，一个合法的括号序列可以有以下三种形式：

- 1 $S = ""$ （空串）， S 是一个合法的括号序列；
- 2 $S = XY$ ，其中 X, Y 均为合法的括号序列，则 S 也是一个合法的括号序列；
- 3 $S = (X)$ ，其中 X 为合法的括号序列，则 S 也是一个合法的括号序列。

这时老师在黑板上写出了一个了括号序列：“()())()”。

Felix 一眼就看出这个序列并不是合法的括号序列。

这时老师提出了一个这样的问题：能否在序列中找出连续的一段，把这一段里面的左括号变成右括号，右括号变成左括号，变换之后整个序列可以变成合法的呢？

Felix 想到，可以把[3..5]进行调换，这样序列就会变为()(())，是一个合法的序列。

很明显，不止有一种方法可以使整个序列变合法。

这时，老师又在黑板上写出了一个长度为 N 的括号序列。Felix 想，能否对这个序列进行至多一次变换，使它变合法呢？

【输入格式】 brackets.in

第一行一个整数 T ，代表数据的组数；接下来 T 行，每一行一组数据。

每组数据一行，代表给出的括号序列。

【输出格式】 brackets.out

输出共 T 行，对于每组数据，输出“possible”（可以变换）或“impossible”（不可变换）。（不含引号）

【样例输入】

3
(
)
)
(
)

【样例输出】

possible
impossible
possible

【数据范围与约束】

对于 50%的数据， $T \leq 5$, $N \leq 20$;
对于 100%的数据， $T \leq 10$, $N \leq 5000$.