

## Задача А. Скобки

Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Требуется определить, является ли правильной данная последовательность круглых скобок. Скобочная последовательность называется правильной, если имеется такое взаимно-однозначное соответствие между открывающими и закрывающими скобками, что подпоследовательность между скобками в каждой паре тоже является правильной скобочной последовательностью.

### Формат входных данных

Первая строка содержит последовательность скобок. Длина строки не превышает 100000 символов.

### Формат выходных данных

Если последовательность является правильной, выведите `CORRECT`. Иначе выведите `INCORRECT`

### Примеры

brackets.in	brackets.out
()(((())())())	CORRECT
()()())()	INCORRECT

## Задача В. Спички

Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 Мб

Есть  $N$  спичек. Необходимо найти максимально возможную площадь прямоугольника, который можно сложить из этих спичек. При построении прямоугольника нельзя ломать спички и не обязательно использовать все спички. Длины всех спичек одинаковы и равны 1 см.

### Формат входных данных

Первая строка содержит целое число  $N$  ( $4 \leq N \leq 1000$ ).

### Формат выходных данных

Выведите максимальную площадь прямоугольника в см<sup>2</sup>.

### Примеры

match.in	match.out
4	1
16	16
18	20
599	22350

## Задача С. Подмножества

Ограничение по времени: 2 секунды  
Ограничение по памяти: 64 мегабайта

Напишите программу, печатающую все подмножества множества  $\{1, 2, \dots, n\}$ , исключая пустое.

### Формат входных данных

Дано натуральное число  $N$  ( $0 < N < 11$ ).

### Формат выходных данных

В выходной файл в каждой строке вывести сначала количество чисел в подмножестве, а затем сами эти числа. Выводить подмножества можно в любом порядке, в каждом подмножестве числа должны быть упорядочены по возрастанию.

### Примеры

subset.in	subset.out
2	2 1 2 1 1 1 2

## Задача D. Баржа

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

На барже располагается  $K$  грузовых отсеков. В каждый отсек можно поместить некоторое количество бочек с одним из 10000 видов топлива. Причём извлечь бочку из отсека можно лишь в случае, если все бочки, помещённые в этот отсек после неё, уже были извлечены. Таким образом в каждый момент времени в каждом непустом отсеке имеется ровно одна бочка, которую можно извлечь не трогая остальных. Будем называть такие бочки крайними.

Изначально баржа пуста. Затем она последовательно проплывает через  $N$  доков, причём в каждом доке на баржу либо погружается бочка с некоторым видом топлива в некоторый отсек, либо выгружается крайняя бочка из некоторого отсека. Однако, если указанный отсек пуст, либо если выгруженная бочка содержит не тот вид топлива, который ожидалось, следует зафиксировать ошибку. Если на баржу оказывается погружено более  $P$  бочек или если после прохождения всех доков она не стала пуста, следует также зафиксировать ошибку. От вас требуется либо указать максимальное количество бочек, которые одновременно пребывали на барже либо зафиксировать ошибку.

### Формат входных данных

В первой строке три целых числа  $N$ ,  $K$  и  $P$  ( $1 \leq N, K, P \leq 100$ ). Далее следует  $N$  строк с описанием действия, выполняемого в очередном доке. Если в нём происходит погрузка, то строка имеет вид «+  $A B$ », где  $A$  — номер отсека, в который помещается бочка, а  $B$  — номер вида топлива в ней. Если же док занимается разгрузкой, то строка имеет вид «-  $A B$ », где  $A$  — номер отсека, из которого извлекается бочка, а  $B$  — номер ожидаемого вида топлива.

### Формат выходных данных

Вывести либо одно число, равное искомому максимуму в случае безошибочного прохождения баржей маршрута, либо вывести слово «Error» в противном случае.

### Пример

ship.in	ship.out
6 1 2	2
+ 1 1	
+ 1 2	
- 1 2	
- 1 1	
+ 1 3	
- 1 3	

## Задача Е. Двоично-единичная сортировка

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Дан набор натуральных чисел. Требуется упорядочить эти числа по следующим странным правилам:

- числа, в двоичной записи которых единиц меньше, должны идти раньше,
- если в двоичной записи двух чисел одинаковое число единиц, то раньше должно идти то из них, которое меньше.

Напишите программу, которая бы, получив на вход набор чисел, упорядочивала эти числа в соответствии с этими правилами.

### Формат входных данных

Первая строка содержит целое число  $N$  ( $1 \leq N \leq 20000$ ). Далее идет  $N$  чисел (каждое число не превышает  $2 \cdot 10^9$ ).

### Формат выходных данных

Выведите выходные числа, упорядоченные по описанным правилам.

### Примеры

<code>binsort.in</code>	<code>binsort.out</code>
3 3 4 5	4 3 5

## Задача F. Трёхцветная улица

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 Мб

Жители Трёхцветной улицы решили раскрасить свои дома, согласно названию, в три цвета — красный, зелёный и синий, каждый в свой цвет. При этом из эстетических соображений жители не хотят, чтобы какие-либо два стоящих рядом дома были покрашены в один и тот же цвет. Для дома номер  $i$  стоящими рядом с ним считаются дома  $i - 1$  и  $i + 1$ ; первый и последний дома имеют по одному соседнему дому.

Недавно состоялось собрание жильцов всех домов этой улицы, на котором было выяснено, во сколько обойдётся покраска каждого дома в каждый из трёх возможных цветов. Теперь жильцы обратились за помощью к Вам, чтобы Вы, располагая этой информацией, выбрали, в какой цвет красить каждый дом так, чтобы учесть эстетические соображения жильцов и при этом заплатить за покраску как можно меньшую сумму.

### Формат входных данных

В первой строке входного файла находится целое число  $N$  ( $1 \leq N \leq 20$ ) — количество домов на Трёхцветной улице. В последующих  $N$  строках записано в каждой по три целых числа  $R_i G_i B_i$  через пробел — стоимость покраски  $i$ -го дома в красный, зелёный и синий цвета, соответственно. Известно, что  $1 \leq R_i, G_i, B_i \leq 1000$ .

### Формат выходных данных

Выведите в первую строку выходного файла одно число — минимальную стоимость требуемой покраски.

### Примеры

rgb.in	rgb.out
3 1 100 100 100 1 100 100 100 1	3
3 1 100 100 100 100 100 1 100 100	102
3 26 40 83 49 60 57 13 89 99	96
6 30 19 5 64 77 64 15 19 97 4 71 57 90 86 84 93 32 91	208
8 71 39 44 32 83 55 51 37 63 89 29 100 83 58 11 65 13 15 47 25 29 60 66 19	253

В первом примере дешевле всего будет покрасить первый дом в красный, второй — в зелёный и третий — в синий цвет.

Во втором примере первый и третий дома надо красить в красный цвет, а второй — в зелёный или синий.