

## Содержание

<b>10.Base [3/3]</b>	<b>3</b>
Задача 10A. Ядра [0.4 sec (0.8 sec), 256 mb]	3
Задача 10B. Рюкзак [0.5 sec, 256 mb]	4
Задача 10C. Ход конём [0.5 sec, 256 mb]	5
<b>10.Advanced [3/7]</b>	<b>6</b>
Задача 10D. Наилучшее приближение [0.8 sec (1.6 sec), 2 mb (16 mb)]	6
Задача 10E. Шаблоны [0.8 sec (1.6 sec), 2 mb (16 mb)]	7
Задача 10F. Интересное число [0.4 sec (0.8 sec), 64 mb]	8
Задача 10G. Коммивояжёр возвращается! [2 sec (4 sec), 256 mb]	9
Задача 10H. Свертка [0.5 sec, 256 mb]	10
Задача 10I. Почтовые отделения [1 sec (2 sec), 256 mb]	11
Задача 10J. Сумма всего подряд [0.5 sec (1 sec), 256 mb]	12
<b>10.Hard [0/2]</b>	<b>13</b>
Задача 10K. Самый длинный путь [0.6 sec (1 sec), 256 mb]	13
Задача 10L. Симпатичные узоры 2 [2 sec (4 sec), 8 mb (32 mb)]	14

**Общая информация:**

Вход в констест: <http://contest.yandex.ru/contest/907/>

Дедлайн на задачи 3-го декабря в 23:59.

К каждой главе есть более простые задачи (base), посложнее (advanced), и сложные (hard).

В скобках к каждой главе написано сколько любых задач из этой главы нужно сдать.

Сайт курса: <http://compscicenter.ru/courses/algorithms-1/2014-autumn/>

Семинары ведет Сергей Владимирович Копелиович,  
контакты: [burunduk30@gmail.com](mailto:burunduk30@gmail.com), [vk.com/burunduk1](https://vk.com/burunduk1)

В каждом условии 2 таймлимита: для C/C++ и для Java, Python.

## 10.Base [3/3]

### Задача 10А. Ядра [0.4 sec (0.8 sec), 256 mb]

Капитан Вася всегда держит на своем корабле запас пушечных ядер для борьбы с пиратами. Так как он привык во всем поддерживать порядок, он хранит ядра в виде пирамид. Каждый из слоев одной пирамиды является равносторонним заполненным ядрами треугольником, сторона которого содержит ровно  $k$  ядер. Сторона основания пирамиды состоит из  $n$  ядер, в следующем слое сторона состоит из  $n - 1$  ядра, и т.д., пока на вершину не будет положено одно ядро (которое является равносторонним треугольником со стороной 1).

Например, пирамида размера 3 состоит из трех уровней, выглядящих так (сверху вниз):

```

X

X
X X

X
X X
X X X
```

Ясно, что каждый из треугольников может содержать только 1, 3, 6, 10 и т.д. ядер. Таким образом, пирамида может содержать только 1, 4, 10, 20, и т.д. ядер.

Вася отправляется в плавание и берет с собой ровно  $m$  ядер. Какое минимальное число пирамид требуется ему сложить из них на своем корабле?

### Формат входных данных

В первой строке входного файла записано количество тестов  $1 \leq T \leq 20$ . В последующих  $T$  строках задается количество ядер в  $i$ -м тесте  $1 \leq m_i \leq 300\,000$ .

### Формат выходных данных

Для каждого из  $T$  тестов входного файла выведите в отдельной строке минимальное количество пирамид.

### Пример

balls.in	balls.out
5	1
1	2
5	3
9	3
15	2
91	

### Задача 10В. Рюкзак [0.5 sec, 256 mb]

Найдите максимальный вес золота, который можно унести в рюкзаке вместительностью  $S$ , если есть  $N$  золотых слитков с заданными весами.

#### Формат входных данных

В первой строке входного файла записаны два числа —  $S$  и  $N$  ( $1 \leq S \leq 10\,000$ ,  $1 \leq N \leq 300$ ).

Далее следует  $N$  неотрицательных целых чисел, не превосходящих 100 000 — веса слитков.

#### Формат выходных данных

Выведите искомый максимальный вес.

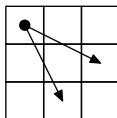
#### Примеры

knapsack.in	knapsack.out
10 3 1 4 8	9
20 4 5 7 12 18	19

### Задача 10С. Ход конём [0.5 sec, 256 mb]

Дана прямоугольная доска  $N \times M$  ( $N$  строк и  $M$  столбцов). В левом верхнем углу находится шахматный конь, которого необходимо переместить в правый нижний угол доски.

При этом конь может ходить следующим образом:



Необходимо определить, сколько существует различных маршрутов, ведущих из левого верхнего в правый нижний угол.

#### Формат входных данных

В первой строке входного файла находятся два натуральных числа  $N$  и  $M$  ( $1 \leq N, M \leq 50$ ).

#### Формат выходных данных

В выходной файл выведите единственное число — количество способов добраться конём до правого нижнего угла доски.

#### Пример

knight.in	knight.out
3 2	1
31 34	293930

## 10.Advanced [3/7]

### Задача 10D. Наилучшее приближение [0.8 sec (1.6 sec), 2 mb (16 mb)]

Вам даны  $N$  целых чисел. Ваша задача — вставить ровно по одному знаку “+” или “-” между каждой парой соседних таким образом, чтобы сделать значение получившегося выражения максимально близким к заданному числу  $A$ .

#### Формат входных данных

Первая строка входного файла содержит два целых числа:  $N$  ( $1 \leq N \leq 10\,000$ ) и  $A$ , которое по модулю не превосходит  $10\,000$ . Далее следуют  $N$  строк, в каждой из которых содержится ровно одно целое число  $X_i$ , не превосходящее по модулю  $10\,000$ . Кроме того, гарантируется, что сумма абсолютных величин всех  $N$  чисел также не превосходит  $10\,000$ .

#### Формат выходных данных

В первой строке необходимо вывести значение получившегося выражения (которое должно быть настолько близко к  $A$ , насколько это возможно). Во второй строке необходимо вывести само выражение, дающее такое значение, в форме  $X_1[+|-]X_2[+|-]\dots X_{N-1}[+|-]X_N$ . Если оптимальных решений несколько, то разрешается выводить любое из них.

#### Пример

nearest.in	nearest.out
3 0	0
3	3+-2-1
-2	
1	

### Задача 10Е. Шаблоны [0.8 sec (1.6 sec), 2 mb (16 mb)]

Многие операционные системы используют шаблоны для ссылки на группы объектов: файлов, пользователей, и т. д. Ваша задача — реализовать простейший алгоритм проверки шаблонов для имен файлов.

В этой задаче алфавит состоит из маленьких букв английского алфавита и точки ('.'). Шаблоны могут содержать произвольные символы алфавита, а также два специальных символа: '?' и '\*'. Знак вопроса ('?') соответствует ровно одному произвольному символу. Звездочка '\*' соответствует подстроке произвольной длины (возможно, нулевой). Символы алфавита, встречающиеся в шаблоне, отображаются на ровно один такой же символ в проверяемой строке. Строка считается подходящей под шаблон, если символы шаблона можно последовательно отобразить на символы строки таким образом, как описано выше. Например, строки "ab", "aab" и "beda." подходят под шаблон "\*a?", а строки "bebe", "a" и "ba" — нет.

### Формат входных данных

Первая строка входного файла определяет шаблон  $P$ . Вторая строка  $S$  состоит только из символов алфавита. Ее необходимо проверить на соответствие шаблону. Длины обеих строк не превосходят 10 000. Строки могут быть пустыми — будьте внимательны!

### Формат выходных данных

Если данная строка подходит под шаблон, выведите YES. Иначе выведите NO.

### Примеры

patterns.in	patterns.out
k?t*n kitten	YES
k?t?n kitten	NO

**Задача 10F. Интересное число [0.4 sec (0.8 sec), 64 mb]**

Для заданного числа  $n$  найдите наименьшее положительное целое число с суммой цифр  $n$ , которое делится на  $n$ .

**Формат входных данных**

Во входном файле содержатся целое число  $n$  ( $1 \leq n \leq 1000$ ).

**Формат выходных данных**

Выходной файл должен содержать искомое число. Ведущие нули выводить не разрешается.

**Пример**

number.in	number.out
1	1
10	190



### Задача 10G. Коммивояжёр возвращается! [2 sec (4 sec), 256 mb]

Коммивояжёр возвращается в систему Альфы Центавра! Население системы с нетерпением ждёт его прибытия — каждый хочет приобрести что-нибудь с далёких планет!

Как обычно, коммивояжёр хочет минимизировать транспортные расходы. Он выбирает начальную планету, прилетает туда на межгалактическом корабле, после чего посещает все остальные планеты системы в порядке, минимизирующем суммарную стоимость посещения, и на другом межгалактическом корабле улетает обратно. Естественно, коммивояжёр не хочет летать ни на какую планету дважды.

Найдите оптимальный маршрут для коммивояжёра. Массы больше не могут ждать!

#### Формат входных данных

В системе Альфы Центавра  $n$  планет. Это число записано в первой строке входного файла ( $1 \leq n \leq 19$ ). Следующие  $n$  строк содержат по  $n$  чисел каждая:  $j$ -ое число на  $i$ -ой из этих строк — стоимость перемещения  $a_{ij}$  от  $i$ -ой планеты до  $j$ -ой. Числа в каждой строке разделены пробелами. Числа  $a_{ii}$  не несут полезной информации. Все числа во входном файле положительны и не превосходят  $10^8$ .

#### Формат выходных данных

В первой строке выходного файла выведите минимальную суммарную стоимость посещения всех планет. Во второй строке выведите  $n$  чисел через пробел — номера планет системы в порядке их посещения. Если оптимальных маршрутов несколько, можно вывести любой из них.

#### Пример

salesman.in	salesman.out
3	5
8 1 6	3 1 2
3 5 7	
4 9 2	

### Задача 10Н. Свертка [0.5 сек, 256 mb]

Петя хочет сократить запись последовательности, состоящей из заглавных латинских букв. Для этого он может свернуть ее повторяющиеся подпоследовательности. Например, последовательность `AAAAAAAAABABABCCD` может быть записана как `10(A)2(BA)B2(C)D`.

Формальное определение свернутой последовательности и соответствующей ей операции развертки дается следующим образом:

- Последовательность, которая содержит единственный символ от 'A' до 'Z' представляет из себя свернутую последовательность. При развертке такой последовательности получается она сама.
- Если  $S$  и  $Q$  — свернутые последовательности, то  $SQ$  также свернутая последовательность. Если при развертке строки  $S$  получается строка  $S'$ , а при развертке  $Q$  получается  $Q'$ , то при развертке  $SQ$  получается строка  $S'Q'$ .
- Если  $S$  — свернутая последовательность, то  $X(S)$  также свернутая последовательность, где  $X$  это десятичное представление целого числа большего единицы. Если при развертке строки  $S$  получается строка  $S'$ , то при развертке  $X(S)$  получается строка  $S'$ , повторенная  $X$  раз.

Петя хочет свернуть заданную последовательность таким образом, чтобы результат содержал наименьшее число символов.

### Формат входных данных

Входной файл содержит непустую строку, состоящую из заглавных латинских букв. Длина строки не превышает 100 символов.

### Формат выходных данных

В выходной файл выведите одну строку, содержащую наименьшую последовательность развертка которой даст строку, заданную во входном файле.

Если ответов несколько — выведите любой из них.

### Пример

folding.in	folding.out
AAAAAAAAABABABCCD	9(A)3(AB)CCD
NEERCYESYESYESNEERCYESYESYES	2(NEERC3(YES))

### Задача 10I. Почтовые отделения [1 sec (2 sec), 256 mb]

Вдоль прямой дороги расположены деревни. Дорога представляется целочисленной осью, а расположение каждой деревни задается одним целым числом — координатой на этой оси. Никакие две деревни не имеют одинаковых координат. Расстояние между двумя деревнями вычисляется как модуль разности их координат.

В некоторых, не обязательно во всех, деревнях будут построены почтовые отделения. Деревня и расположенное в ней почтовое отделение имеют одинаковые координаты. Почтовые отделения необходимо расположить в деревнях таким образом, чтобы общая сумма расстояний от каждой деревни до ближайшего к ней почтового отделения была минимальной.

#### Формат входных данных

В первой строке содержатся два целых числа: количество деревень  $n$  ( $1 \leq n \leq 300$ ) и количество почтовых отделений  $m$  ( $1 \leq m \leq 30$ ),  $m \leq n$ . Вторая строка содержит  $n$  целых чисел в возрастающем порядке, являющихся координатами деревень. Для каждой координаты  $x$  верно  $1 \leq x \leq 10^4$ .

#### Формат выходных данных

Первая строка выходного файла должна содержать одно целое число — общую сумму расстояний от каждой деревни до её ближайшего почтового отделения. Вторая строка должна содержать  $m$  целых чисел в возрастающем порядке. Эти числа являются искомыми координатами почтовых отделений. Если для заданного расположения деревень есть несколько решений, необходимо найти любое из них.

#### Пример

post.in	post.out
10 5	9
1 2 3 6 7 9 11 22 44 50	2 7 22 44 50

### Задача 10J. Сумма всего подряд [0.5 sec (1 sec), 256 mb]

Дан случайный граф. Нужно для каждого множества вершин  $A$  посчитать  $f(A)$ , количество независимых подмножеств вершин  $B$ :  $B \subseteq A$ . Множество вершин  $B$  называется независимым, если в графе нет ребра, оба конца которого лежат в множестве  $B$ .

#### Формат входных данных

На первой строке число вершин  $n \geq 1$  и число ребер  $m \geq 1$ .

Следующие  $m$  строк содержат пары чисел от 1 до  $n$  — ребра графа.

В графе нет ни петель, ни кратных ребер.

#### Формат выходных данных

Каждому множеству  $A$  можно сопоставить целое число  $b(A)$ , двоичная запись которого соответствует наличию элементов в множестве  $A$ . Пример:  $n = 5, A = \{1, 2, 5\}, b(A) = 2^0 + 2^1 + 2^4 = 19$ . Выведите  $\sum_A f(A) 2^{b(A)} \bmod (10^9 + 7)$

#### Система оценки

Подзадача 1 (25 баллов)  $1 \leq n \leq 10$ .

Подзадача 2 (25 баллов)  $1 \leq n \leq 16$ .

Подзадача 3 (30 баллов)  $1 \leq n \leq 20$ .

Подзадача 4 (20 баллов)  $1 \leq n \leq 23$ .

#### Примеры

subsetsum.in	subsetsum.out
3 1 1 2	1221

#### Пояснение к примеру

Независимыми являются множества вершин  $\{\}, \{1\}, \{2\}, \{3\}, \{1, 3\}, \{2, 3\}$ .

$A = \{\}$   $f(A) = 1$   $b(A) = 0$

$A = \{1\}$   $f(A) = 2$   $b(A) = 2^0 = 1$

$A = \{2\}$   $f(A) = 2$   $b(A) = 2^1 = 2$

$A = \{1, 2\}$   $f(A) = 3$   $b(A) = 2^0 + 2^1 = 3$

$A = \{3\}$   $f(A) = 2$   $b(A) = 2^2 = 4$

$A = \{1, 3\}$   $f(A) = 4$   $b(A) = 2^0 + 2^2 = 5$

$A = \{2, 3\}$   $f(A) = 4$   $b(A) = 2^1 + 2^2 = 6$

$A = \{1, 2, 3\}$   $f(A) = 6$   $b(A) = 2^0 + 2^1 + 2^2 = 7$

$$1 \cdot 2^0 + 2 \cdot 2^1 + 2 \cdot 2^2 + 3 \cdot 2^3 + 2 \cdot 2^4 + 4 \cdot 2^5 + 4 \cdot 2^6 + 6 \cdot 2^7 = 1221$$

## 10.Hard [0/2]

### Задача 10K. Самый длинный путь [0.6 sec (1 sec), 256 mb]

В данном ориентированном графе найдите самый длинный путь такой, что каждая вершина графа встречается в нём не более одного раза.

#### Формат входных данных

В первой строке входного файла заданы через пробел два целых числа  $n$  и  $m$  ( $1 \leq n \leq 22$ ,  $0 \leq m \leq 1000$ ). В следующих  $m$  строках заданы рёбра графа в формате  $u_i \ v_i$  — номера начальной и конечной вершин ребра  $i$ , соответственно. Граф может содержать петли и кратные рёбра.

#### Формат выходных данных

В первой строке выходного файла выведите длину искомого пути  $l$ . Во второй строке выведите  $l + 1$  число через пробел — вершины пути в порядке обхода. Если оптимальных ответов несколько, можно вывести любой из них.

#### Примеры

path.in	path.out
3 3 1 2 2 3 3 1	2 1 2 3
4 6 1 2 2 1 2 3 2 4 3 2 4 2	2 1 2 4
5 3 3 2 2 2 1 5	1 3 2

### Задача 10L. Симпатичные узоры 2 [2 sec (4 sec), 8 mb (32 mb)]

Компания *BrokenTiles* планирует заняться выкладыванием во дворах у состоятельных клиентов узор из черных и белых плиток, каждая из которых имеет размер  $1 \times 1$  метр. Известно, что дворы всех состоятельных людей имеют наиболее модную на сегодня форму прямоугольника  $n \times m$  метров.

Однако при составлении финансового плана у директора этой организации появилось целых две серьезных проблемы: во первых, каждый новый клиент очевидно захочет, чтобы узор, выложенный у него во дворе, отличался от узоров всех остальных клиентов этой фирмы, а во вторых, этот узор должен быть симпатичным.

Как показало исследование, узор является симпатичным, если в нем нигде не встречается квадрата  $2 \times 2$  метра, полностью покрытого плитками одного цвета.

Для составления финансового плана директору необходимо узнать, сколько клиентов он сможет обслужить, прежде чем симпатичные узоры данного размера закончатся. Помогите ему!

#### Формат входных данных

На первой строке входного файла находятся два натуральных числа  $n$  и  $m$ .  $1 \leq n \cdot m \leq 300$ .

#### Формат выходных данных

Выведите в выходной файл единственное число — количество различных симпатичных узоров, которые можно выложить во дворе размера  $n \times m$  по модулю  $2^{30} + 1$ . Узоры, получающиеся друг из друга сдвигом, поворотом или отражением считаются различными.

#### Пример

nice2.in	nice2.out
2 2	14
3 3	322