



## Задача А. “Транслитерация”

Еще будучи школьником Хрусталькин понимал, что русские буквы не пользуются популярностью среди программистов. Но он не думал, что проблема возникнет с записью русских имен и фамилий. Профессор Питонов требовал от студентов все папки и файлы с выполненными работами именовать с использованием букв латинского алфавита, причем в соответствии с установленными профессором правилами. Если папка или файл были именованы не по правилам, профессор даже не смотрел работу.

Хрусталькин решил написать программу, которая преобразует русский текст с использованием символов латинского алфавита (у программистов такое преобразование называется *транслитерация*) по выдвинутым профессором правилам.

### 1. Правила замены букв:

А – A	Ё – E	Л – L	С – S	Ч – CH	Я – IA
Б – B	Ж – ZH	М – M	Т – T	Ш – SH	
В – V	З – Z	Н – N	У – U	Щ – SHCH	
Г – G	И – I	О – O	Ф – F	Ы – Y	
Д – D	Й – I	П – P	Х – KH	Э – E	
Е – E	К – K	Р – R	Ц – TC	Ю – IU	

2. Букву «ё» транслитерируйте как «e», «й» как «i», а «ъ» и «ь» (и их заглавные версии «Ъ» и «Ь») должны исчезнуть из текста.
3. Строчные буквы заменяются на строчные, заглавные заменяются на заглавные.
4. Если заглавная буква превращается при транслитерации в несколько букв, то заглавной должна остаться только первая из них (например, «Ц» → «TC»).
5. Все некириллические символы должны остаться на месте.

Программа, написанная Хрусталькиным, успешно справлялась с переводом. А получится ли у вас написать аналогичную программу?

### Входные данные

В единственной строке задан русский текст. Текст может состоять из любых символов. Вам необходимо транслитерировать только русские буквы, а остальные оставить на месте. Гарантируется, что нет слов, состоящих только из букв «ъ» и «ь».

### Выходные данные

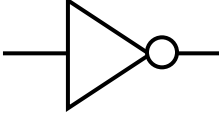
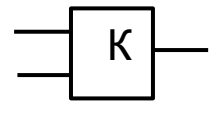
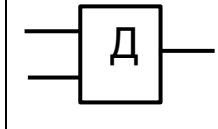
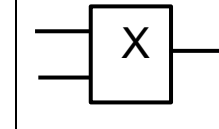
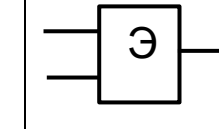
Выведите одну строку – транслитерированный текст.

### Примеры

Входная строка	Результат
Чудес не бывает!	Chudes ne byvaet!
Скоро лето, ЕГЭ, поступление в вуз.	Skoro leto, EGE, postuplenie v vuz.

**Задача В. “Логическая ошибка”**

Первокурсник Свистулькин на лабораторной работе по «Основам построения цифровых логических устройств», собрал схему, реализующую некоторую логическую функцию  $F(A, B, C)$ , используя имеющиеся в наборе элементы следующих типов (в наборе по 10 элементов каждого типа):

Обозначение элемента					
Логическая операция	Отрицание (NOT)	Конъюнкция (AND)	Дизъюнкция (OR)	Исключающее ИЛИ (XOR)	Эквиваленция ( $\leftrightarrow$ )
Стоимость элемента, у.е.	35	40	30	50	45

Собранная Свистулькиным схема имела вид (рис. 1):

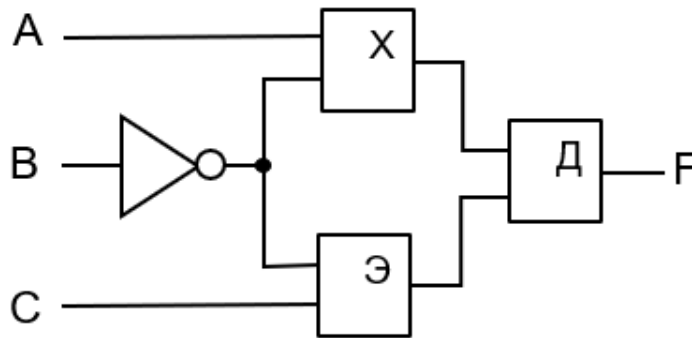


Рисунок 1. Схема с ошибкой, собранная Свистулькиным

Протестировав собранную схему, профессор Винтиков сказал, что Свистулькин ошибся, а схема должна для каждой комбинации входных значений ( $A, B, C$ ) выдавать выходное значение  $F$  ровно **противоположное** тому, которое получается для этой комбинации по построенной студентом схеме.

К сожалению, все связи и элемент отрицания Свистулькин прочно припаял, но ему удалось извлечь три оставшихся элемента (обозначенные на рис. 1 буквами Х, Э, Д). Теперь он должен исправить свою работу, поставив на освободившееся место (на рис. 2 позиции 1, 2, 3) любой элемент из набора.

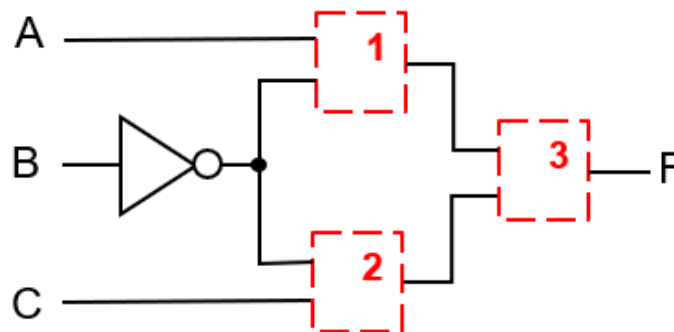


Рисунок 2. Схема после извлечения элементов с позиций 1, 2, 3

Свистулькину удалось подобрать нужные элементы и поставить их на соответствующие позиции в схеме.



Схема, собранная Свистулькиным, является частью целой установки, которую он делал совместно с профессором для участия в проекте «Россия – страна возможностей». Схема всей установки реализует некоторую логическую функцию  $G(A, B, C)$  и имеет вид (рис. 3):

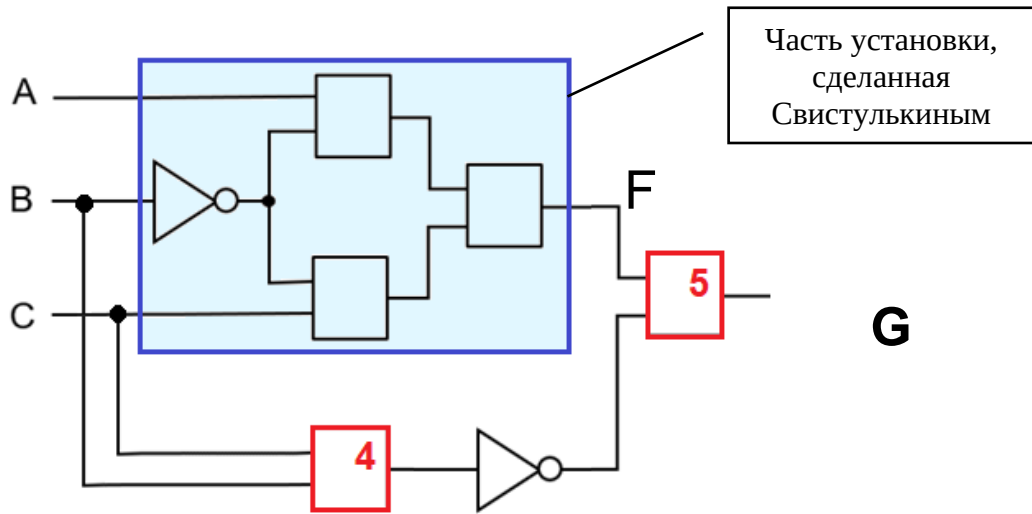


Рисунок 3. Схема всей установки, реализующая логическую функцию  $G$

Для работы всей установки профессор подбирал элементы из имеющегося набора в позиции 4 и 5, таким образом, чтобы они реализовывали разные логические функции, а на выходе  $G$  всей установки получалось значение *FALSE* только в случае равенства между собой всех входных значений ( $A, B, C$ ), в остальных случаях на выходе установки должно получаться значение *TRUE*, причем стоимость всей установки должна быть минимальна.

Определите, какие логические элементы подобрал профессор и какова при этом оказалась стоимость всей установки.

### Выходные данные

В ответе в разных строках укажите:

- в первой строке – без пробелов запишите заглавные буквы, соответствующие элементам, которые нужно расположить на позициях 1, 2, 3: первой запишите букву, соответствующую позиции 1, затем букву, соответствующую позиции 2 и в конце букву, соответствующую позиции 3;
- во второй строке – без пробелов запишите заглавные буквы, соответствующие элементам, которые нужно расположить на позициях 4, 5;
- в третьей строке – минимальная стоимость всей установки.

### Примечания

Отличие нотации приведенных логических схем от стандартизированной объясняется необходимостью сделать удобным ввод ответов.

### Таблица для заполнения

Вопрос задачи	Ответ
Элементы, которые нужно расположить на позициях 1, 2, 3	
Элементы, которые нужно расположить на позициях 4, 5	
Минимальная стоимость всей установки	



**Решение:**



## Задача С. “Ностальгия по ЕГЭ”

Старшеклассник Буквоедов готовится сдавать ЕГЭ по информатике и часто, решая задачи по программированию, обращается за помощью к своему соседу – первокурснику Рязанцеву. Рязанцев раньше считал, что ЕГЭ – самое страшное, что в жизни могло с ним произойти, но после лекций и экзамена по высшей математике передумал и вспоминает те далекие времена с ностальгией, а задачи ЕГЭ решает теперь с удовольствием.

Вот и сейчас Буквоедов просит решить очередную задачу. Ему требуется определить сколько существует чисел,  $q$ -ичная ( $10 \leq q \leq 13$ ) запись которых состоит из  $(16 - q)$  цифр и для которых выполняются следующие условия:

- 1) в  $q$ -ичной записи числа обязательно содержится хотя бы один вводимый с клавиатуры символ;
- 2) в  $q$ -ичной записи числа никакие две чётные и две нечётные цифры не стоят рядом;
- 3) среди делителей самого числа есть как минимум два простых двузначных делителя.

Рязанцев написал программу для нахождения нужного количества комбинаций, но в ответе не уверен. Он хочет, чтобы вы ему помогли.

### Входные данные

Во входной строке через пробел записаны:  $q$  – основание системы счисления ( $10 \leq q \leq 13$ ) и символ, который обязательно должен присутствовать в  $q$ -ичной записи числа.

### Выходные данные

В единственной выходной строке необходимо вывести количество чисел, удовлетворяющих условиям задачи.

### Примеры

Стандартный ввод	Стандартный вывод
13 7	11
13 С	14



## Задача D. “Сборщик налогов”

Царь некоторого квадратного государства с целью наведения порядка в финансовом секторе поделил зону своего правления на квадратные области (всего получилось  $25 \times 25$  клеток) и разделил их на цветовые зоны: желтые, голубые и светло-зелёные. Принцип раскраски областей остался никому из подчиненных не понятным, но спорить с Царем никто не стал.

По каждой области есть информация о количестве золотых монет, которые остаются к концу года в ее казне. Перед Новым Годом Царь сам лично объезжал области и забирал в общую казну государства в качестве налога часть золотых монет из казны области, причем действовал по следующим правилам:

- если в казне области было меньше 10 золотых, царь не забирал из казны ничего;
- если он заезжал в «жёлтую» область, то забирал половину золотых из ее казны;
- если он заезжал в «светло-зелёную» область, то забирал треть золотых из ее казны;
- если он заезжал в «голубую» область, то забирал четверть золотых из ее казны.

Царь был не жадным, поэтому, если монеты ровно не делились, он их не распиливал, а брал ближайшее максимальное целое количество монет.

Но после пандемии коронавируса силы стали покидать Царя и собирать налоги ему стало тяжело. Доверить сбор налогов подчиненным он не мог, поскольку они всегда воровали. Советник Царя, Наложкин, окончивший отделение робототехники, посоветовал приобрести роботов, которые не только заменят ручной труд, но еще позволят сэкономить государственные деньги.

Так у Царя появился робот «Налогосборщик», который сам объезжал все области, собирал золото за Царя и выдавал чеки о приеме золотых. Робот исправно служил Царю 10 лет.

Запуская в очередной раз робота за сбором золотых, Царь обнаружил, что механизмы поворота и счетчик монет у него сломались. Наложкин был в отпуске и быстро отремонтировать робота не мог, а поскольку Царю срочно нужны были золотые на выплату зарплаты подчиненным, он решил отправить «Налогосборщика» за золотом, несмотря на поломки.

При тех поломках, которые были у робота он мог перемещаться по государству только на одну государственную клетку **вправо, вниз** или **по диагонали вправо вниз**. Поскольку счетчик монет сломался и выводил робота из строя, как только при пересчете момент их оказывалось больше 100, Царь **запретил ему заезжать в те квадратные области, в казне которых было более 100 золотых**.

Запустив робота из верхней левой клетки своего государства, сам Царь на ковре-самолете отправился в нижнюю правую клетку, где должен был финишировать робот. По дороге, понимая, что робот пропустит очень много областей, Царь прикидывал сколько минимум и максимум золотых сможет при таких проблемах собрать «Налогосборщик». После расчетов царь успокоился, поскольку даже при получении минимального количества золотых их хватало на зарплату подчиненных, а при получении планируемого максимума Царь мог бы купить в подарок к Новому Году шубку своей Царевне.

Определите какие значения собранного минимального и максимального количества золотых получились у Царя. В ответе укажите эти два числа **через пробел: сначала минимум, потом максимум**.

Исходные данные записаны в электронной таблице **Robot.xlsx**, каждая ячейка соответствует клетке квадрата (в ней записано целое неотрицательное число, соответствующее количеству золотых, имеющихся в казне области) и промаркирована цветом.

**Примечание:** Файл с исходными данными «*Robot.xlsx*» находится в папке **C:\Olimp\_2024**.



## Задача Е. “Бабушкино хобби”

Бабушка Свистулькина – большой поклонник игры «Лото». Поэтому, когда внук приезжает к ней на каникулы, часто просит поиграть с ней.

Известно, что во время игры в «Лото» каждому игроку выдается набор из трех карточек лото. На каждой карточке в трех строках и девяти столбцах написаны цифры от 1 до 90. Числа на карточке располагаются в соответствии с некоторыми правилами: в первом столбце стоят числа от 1 до 9, во втором – от 10 до 19, в третьем – от 20 до 29 и т.д.



Свистулькину, выросшему на компьютерных играх-стрелялках и гонках, темп игры не очень нравится. Но, будучи большим поклонником цифр, он придумал себе занятие: во время игры производил манипуляции с цифрами и числами на карточках, выясняя наличие некоторых закономерностей. Его интересовали два вопроса: есть ли у него уникальная карточка и образуют ли три карточки лото идеальный набор.

По мнению Свистулькина:

- **карточка уникальна**, если сумма **цифр** в каждой строке карточки равна одному и тому же числу;
- **набор** из трех карточек лото **идеален**, если на всех трех карточках совпала сумма **чисел**.

Но, как только Свистулькин начинал складывать цифры и числа, то либо быстро сбивался, либо не успевал закрывать числа на карточке фишками и проигрывал бабушке. Он решил написать программу, которая будет проверять уникальность карточки и идеальность всего набора.

### Входные данные

На обработку поступает файл *input.txt*, в котором содержатся 9 строк: строки с номерами 1–3 соответствуют первой карточке набора; строки с номерами 4–6 – второй карточке, строки с номерами 7–9 – третьей карточке (нумерация строк сверху вниз). В каждой строке записано 9 чисел (от 0 до 90), разделенных одним или несколькими пробелами, число ноль в строке означает отсутствие цифры в соответствующем столбце карточки лото. Остальные числа записаны в строке по правилам расположения их по столбцам карточки при игре в «Лото».

### Выходные данные

В первой строке необходимо вывести слово YES, если есть уникальная карточка, затем через пробел указать её номер и значение суммы **цифр**, совпавшее по всем трем строкам этой карты. Если уникальных карточек несколько, то после слова YES надо вывести номера всех уникальных карточек с соответствующим значением суммы, разделив выходные данные одним пробелом в формате:

```
YES номер сумма номер сумма номер сумма
```

Номера карточек должны идти по возрастанию. Если в наборе нет уникальной карточки, в первой строке вывести слово NO.



Во второй строке необходимо вывести слово YES, если набор из трех карточек оказался идеальным, затем через пробел указать значение суммы **чисел**, совпавшее на всех трех карточках. Если набор оказался неидеальным, во второй строке вывести слово NO, а затем через пробел вывести сумму чисел, записанных на первой, второй и третьей карточке.

### Примеры

Содержимое входного файла <i>input.txt</i>	Стандартный вывод
3 0 25 38 42 0 69 0 0 0 18 28 0 41 51 0 0 84 6 0 0 30 0 0 67 73 82 4 0 29 0 0 0 62 74 83 3 11 0 0 49 55 0 0 89 0 14 25 37 0 59 63 0 0 8 12 0 0 40 0 64 0 81 4 19 0 32 42 0 0 77 0 0 0 24 39 0 52 0 75 88	YES 1 42 2 45 YES 657
2 13 0 39 0 57 0 79 0 0 0 20 0 46 54 68 0 0 4 17 0 0 42 0 0 76 84 0 12 0 37 0 56 68 75 0 0 0 23 0 0 50 0 76 84 7 19 21 34 45 0 0 0 0 3 0 27 35 0 0 67 72 0 0 12 28 0 46 50 0 77 0 0 15 0 34 0 56 64 0 80	YES 3 42 NO 601 607 666





## Задача F. “Буквы, стройся!”

Тема анализа контента является в настоящее время востребованной. **Контент-анализ** – это анализ формы и содержания текстов и информационных источников: их обработка, оценка и интерпретация. Он позволяет исследовать объекты, явления, процессы и свойства социальной реальности, которые представлены в этих текстах. Анализировать можно любые источники информации, например, новости в СМИ на определённую тему, учебники, архивные документы, исследовательские интервью, программы политических партий, посты в социальных сетях и т.д. Тема эта довольно интересна и многогранна. Контент-анализ позволяет учитывать, в частности, упоминание определённых элементов текста и частоту их появления.

Первокурсник Коломенцев только начал заниматься вопросами контент-анализа. Ему требуется проанализировать предлагаемый текст.

Суть задачи в следующем: дан текстовый файл *input.txt*, состоящий из заглавных латинских букв (**A..Z**), пробелов и цифр от **0** до **9** (всего в файле не более чем  $10^6$  символов). Текст в файле разбит на строки различной длины. Необходимо найти строку, содержащую самое большое количество чисел, в записи которых все цифры различны (если таких строк несколько, надо взять ту, которая в файле встретилась позже). Под **числом** понимаем последовательность записанных подряд цифр без ведущих нулей, между которыми нет букв и пробела. В найденной строке надо проанализировать частоту вхождения в нее букв и вывести в обратном алфавитном порядке без пробелов те буквы, которые в строке встречаются более 3-х раз. Если в найденной строке нет букв, которые встречаются более 3-х раз, то надо вывести самое большое число из найденной строки, в котором все цифры различны. Если в файле нет строки, содержащей число, в записи которого все цифры различны, выведите слово **NO**.

### Входные данные

На обработку поступает файл *input.txt*, состоящий из заглавных латинских букв (**A..Z**), пробелов и цифр от **0** до **9** (всего в файле не более чем  $10^6$  символов).

### Выходные данные

В выходной строке без пробелов выведите в обратном алфавитном порядке буквы, которые встречаются более 3-х раз в строке, содержащей самое большое количество чисел с различными цифрами. Если в найденной строке нет букв, которые встречаются более 3-х раз, то надо вывести самое большое число из найденной строки, в котором все цифры различны. Если в файле не оказалось строки, содержащей число, в записи которого все цифры различны, вывести **NO**.

### Примеры

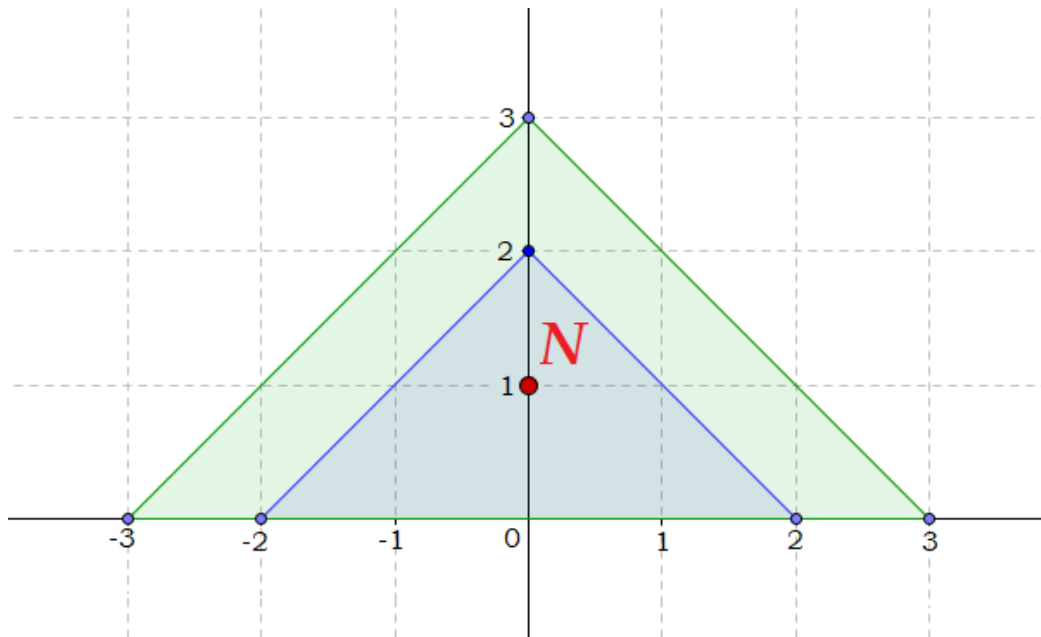
Содержимое входного файла <i>input.txt</i>	Стандартный вывод
AS22E GF 3H F137FG ASW13M AWTMA199RWM FGBH941NJG145 7BJFFJH DA4F81DA317DAFDM FM565 MDAM DF20MM 343RF M97MDM	MDA
DADA34 23 HGFJ678 TR FGHD 234 DDR 759AN SSDGLBX 123X781 45RX	781



## Задача G. “Треугольные страны”

Эта история происходила на одной плоской планете. С незапамятных времен на ней существовал город  $N$ , находящийся в точке  $(x_n, y_n)$ . Кроме этого, в разное время на этой же планете существовали страны, каждая из которых имела форму треугольника.

Теперь перед историками встала серьезная задача – по имеющимся у них данным о треугольных странах определить, в какие страны мог входить город  $N$ . Город мог входить в страну, если он находился строго внутри нее.



### Входные данные

В первой строке записаны через пробел два целых числа:  $x_n$  и  $y_n$  – координаты города  $N$ . Во второй строке вводится количество  $k$  треугольных стран ( $0 \leq k \leq 1000$ ). Каждая из последующих  $k$  строк описывают одну треугольную страну. Описание треугольной страны состоит из шести целых чисел  $x_1, y_1, x_2, y_2, x_3, y_3$ , где  $(x_1, y_1)$ ,  $(x_2, y_2)$ ,  $(x_3, y_3)$  – координаты вершин этой страны.

Гарантируется, что все страны имеют ненулевую площадь. Все координаты не превосходят 10000 по абсолютной величине.

### Выходные данные

В первой строке выведите количество стран, в которые мог входить город  $N$ . Во второй строке выведите через пробел номера этих стран в возрастающем порядке. Страны нумеруются с единицы в том порядке, в каком они заданы во входном файле.

### Примеры

Стандартный ввод	Стандартный вывод
<pre>0 1 2 -2 0 2 0 0 2 -3 0 3 0 0 3</pre>	<pre>2 1 2</pre>
<pre>0 2 2 -2 0 2 0 0 2 -3 0 3 0 0 3</pre>	<pre>1 2</pre>



## Задача Н. “Эффективный расход памяти”

Для участия в конкурсе «Россия – страна возможностей» одnogруппники Хрусталькин, Рязанцев и Коломенцев решили написать программный комплекс для мобильных устройств с элементами искусственного интеллекта, который при телефонном разговоре с абонентом, отсутствующим в списке контактов, мог обнаружить факт мошенничества и предупредить владельца о последствиях разговора. Этот комплекс позволил бы поменять жизнь людей к лучшему, а разработчикам развивать профессиональные компетенции, поэтому у ребят был большой шанс выиграть в конкурсе. По условиям конкурса на разрабатываемые программные комплексы накладывалось ограничение по объёму занимаемой памяти.

В качестве языка программирования был выбран Python, поскольку студенты хорошо его знали, и в нем уже имелось большое количество библиотек (модулей), содержащих некоторую совокупность полезных процедур и функций, необходимых для реализации комплекса. Но, к сожалению, после импорта всех необходимых для работоспособности комплекса модулей оказалось, что его размер превысил установленный лимит, что огорчило студентов. Изучив документацию по импортируемым библиотекам и их состав, студенты выяснили, что содержимое некоторых модулей могло частично пересекаться с содержимым других модулей. То есть, одну и ту же функцию (процедуру) можно взять из различных модулей.

Каждый модуль характеризуется двумя параметрами: **эффективностью** (быстродействием) и **объёмом** занимаемой памяти. Теперь студентам требуется решить вопрос об импорте нужных модулей таким образом, чтобы суммарный объём памяти импортируемых модулей не превысил заданного размера, а суммарная их эффективность была максимально возможной.

### Входные данные

На обработку поступает файл *input.txt*. Первая строка входного файла содержит два целых числа:

- $n$  – количество модулей ( $5 \leq n \leq 120$ );
- $v$  – объём доступной памяти ( $10 \leq v \leq 1000$ ).

Следующие  $n$  строк содержат пары целых чисел (разделенных пробелом), определяющих соответственно объём и эффективность каждого модуля. Гарантируется, что хотя бы один модуль помещается в выделенном объёме памяти.

### Выходные данные

В первой строке выведите через пробел максимально возможную суммарную эффективность отобранных модулей и суммарный объём требующейся для них памяти. Во второй строке выведите через пробел номера этих модулей в возрастающем порядке. Модули нумеруются с единицы в том порядке, в каком они заданы во входном файле.

### Примеры

Содержимое входного файла <i>input.txt</i>	Стандартный вывод
3 20 15 19 6 10 7 11	21 13 2 3
5 14 7 22 5 19 10 10 4 17 6 8	41 12 1 2