

Задача А. Путь редактора

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася поставил перед собой грандиозную цель — искоренить опечатки. Он заметил, что в большинстве случаев опечатки можно разделить на три типа:

- одна буква в слове пропущена (например, вместо *пока* пишут *пка*);
- в слово дописана одна лишняя буква (например, вместо *привет* пишут *приветь*);
- одна из букв в слове заменена на другую (например, вместо *корова* пишут *карова*).

Вася хочет провести статистические исследования, и для этого ему нужно знать, сколько существенно различных опечаток можно допустить в конкретном слове — то есть, сколько существует различных слов, которые можно получить, совершив одну опечатку (любого типа) в данном.

Формат входного файла

В первой строке находится единственное число N ($1 \leq N \leq 26$). Исходное слово и слова с опечатками могут содержать только первые N букв алфавита. Во второй строке находится слово S ($2 \leq |S| \leq 100\,000$), состоящее из маленьких латинских букв.

Формат выходного файла

Выведите одно число — количество различных слов, которые можно получить, совершив единственную опечатку в слове S , и которые составлены только из первых N символов алфавита.

Примеры

<i>Вход</i>	<i>Выход</i>
4 abcd	32

Примечания

В примере из строки `abcd` можно получить следующие строки:

```
abc  abd  acd  bcd
aacd abad abbd abca
abcb abcc abdd accd
adcd bbcd cbcd dbcd
aabcd abacd abbcd abcad
abcbd abccd abcda abcdb
abcdc ab added ab dcd acbcd
adbcd babcd cabcd dabcd
```

Задача В. Ломаная

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася очень не любит уроки истории. И вместо того, чтобы слушать интереснейшие рассказы учительницы Марии Федоровны, про то как Иван Грозный брал Казань, Астрахань, Ревель и, при этом, не брал Шпака, он рисует ломаные линии в своей тетради в клеточку. Несмотря на нелюбовь к истории, Вася очень трепетно относится к красоте и порядку. Поэтому звенья ломаной строго параллельны (перпендикулярны) сторонам тетради, а вершины находятся точно в серединах клеток.

Коля, Васин сосед по парте, долго молча наблюдал за процессом рисования ломаных. И вдруг, неожиданно отметил в тетради две клетки и спросил: «Можешь ли ты соединить эти две клетки ломаной, которая бы пересекала нечетное количество сторон клеток, при этом два раза не посещая одну клетку?»

Помогите Васе правильно ответить на вопрос и не ударить в грязь лицом перед товарищем.

Формат входного файла

В первой строке входных данных находятся два целых числа N и M — размеры листа тетради в клетках ($1 \leq N, M \leq 1\,000\,000$).

Во второй строке находятся два целых числа — координаты первой клетки X_1, Y_1 , отмеченной Колей ($1 \leq X_1 \leq N, 1 \leq Y_1 \leq M$).

В третьей строке находятся два целых числа — координаты второй клетки X_2, Y_2 , отмеченной Колей ($1 \leq X_2 \leq N, 1 \leq Y_2 \leq M$).

Гарантируется, что Коля отметил две разные клетки.

Формат выходного файла

В единственной строке выведите слово YES, если ломаная, удовлетворяющая требованиям, существует, или NO — в противном случае.

Примеры

<i>Вход</i>	<i>Выход</i>
2 2 1 1 2 2	NO
5 3 2 2 4 3	YES

Задача С. База данных

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася занимается разработкой собственной базы данных. База данных содержит записи, представляющие собой массивы из M целых чисел. Основным запросом к базе данных является поиск i -ой по счету записи после сортировки записей по убыванию их j -ых элементов. Вася очень хочет, чтобы запросы выполнялись как можно быстрее, поэтому просит Вас о помощи в написании программы.

Формат входного файла

В первой строке находятся два целых числа N и M — количество записей в базе данных и количество чисел в каждом массиве соответственно ($1 \leq N, M \leq 1000$).

В последующих N строках находятся по M чисел $a_{i,j}$ ($1 \leq i \leq N, 1 \leq j \leq M$), описывающих очередную запись в базе данных ($1 \leq a_{i,j} \leq 1\,000\,000\,000$).

Строка $N + 2$ содержит целое число K — количество запросов к базе данных ($1 \leq K \leq 1000$).

Последующие K строк содержат описания запросов — пары чисел (c_i, d_i) ($1 \leq c_i \leq N, 1 \leq d_i \leq M$), обозначающие, что требуется вывести c_i -ую по счету запись после сортировки записей по убыванию их d_i -ых элементов. Записи и их элементы нумеруются с 1.

Гарантируется, что элементы массива, имеющие одинаковый номер, во всех записях различны, т.е. для всех j ($1 \leq j \leq M$) не существует такой пары (p, q) ($1 \leq p, q \leq N$) таких, что $a_{p,j} = a_{q,j}$.

Формат выходного файла

K строк, содержащих по M чисел в каждой — ответы на запросы в порядке их появления во входных данных.

Примеры

<i>Вход</i>	<i>Выход</i>
3 3	11 19 50
10 20 30	13 15 40
11 19 50	11 19 50
13 15 40	
3	
2 2	
1 1	
1 3	

Задача D. Алиби

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася читает захватывающий детективный роман. Как раз сейчас детективы загнали злодея в угол, и, кажется, осталась формальность — доказать виновность подозреваемого в суде. Но у адвоката есть хитрый план. Он знает, что в наше время сложно передвигаться незамеченным и собрал записи с десятков камер наблюдения, установленных на площадях. Зная примерное время совершения преступления и опираясь на записи камер, адвокат утверждает, что подсудимый физически не успел бы добраться до места преступления. Детективам придется потрудиться, чтобы разбить эту защиту. К счастью, им помогает гениальный программист, которому не составит труда написать программу, чтобы проанализировать собранные адвокатом улики.

Детективы уже выяснили, что все данные, которые предоставил адвокат — истинны, то есть подозреваемый действительно находился в указанное время на указанной площади. Тем не менее, эти данные не полны — отсутствие указания, что подозреваемый находился где-то, не значит, что его там не было: адвокат мог не выбрать эту запись или камера не зафиксировала лица подозреваемого.

Формат входного файла

Детективы уже перевели все даты и времена, которые встречаются во входных данных во *временные отметки* — количество минут, прошедших с некоторого фиксированного момента в прошлом (этот момент один для всех входных данных).

В первой строке находятся два числа N, M — количество площадей в городе и количество дорог соответственно ($1 \leq N \leq 1\,000, 0 \leq M \leq \frac{N(N-1)}{2}$).

В следующих M строках описывается карта города. В каждой строке — три числа a_i, b_i, l_i ($1 \leq a_i < b_i \leq N, 1 \leq l_i \leq 10\,000$) — номера площадей, соединенных дорогой и время, нужное для перемещения по этой дороге (в минутах). Каждая пара площадей соединена не более чем одной дорогой. По любой дороге можно двигаться в обе стороны.

В следующей строке записаны три числа C, t_b и t_e ($1 \leq C \leq N, 0 \leq t_b \leq t_e, \leq 15\,000\,000$) — номер площади на которой произошло преступление, самое раннее и самое позднее время, в которые могло быть совершено преступление (в виде временных отметок).

В следующей строке находится одно число K ($1 \leq K \leq 50\,000$) — количество показаний с камер наблюдения, предъявленных адвокатом.

В следующих K строках записаны пары чисел p_j, t_j ($1 \leq p_j \leq N, 0 \leq t_j \leq 15\,000\,000$) — номер площади и время, когда камера зафиксировала подозреваемого (также в виде временной отметки). Все временные отметки t_j различны.

Все числа во входных данных — целые.

Формат выходного файла

Выведите YES, если подозреваемый мог побывать на месте преступления, или NO, если он не мог этого сделать.

Примеры

<i>Вход</i>	<i>Выход</i>
4 6 1 2 10 2 3 10 3 4 10 1 4 10 1 3 15 2 4 15 2 12 25 2 1 10 3 30	YES
4 6 1 2 10 2 3 10 3 4 10 1 4 10 1 3 15 2 4 15 2 12 15 2 1 10 3 25	NO

Примечания

В первом примере подозреваемый мог по пути из 1 в 3 зайти на площадь 2 и оказался бы там в момент времени 20 — как раз вовремя для совершения преступления.

Во втором примере подозреваемый никак не может успеть с площади 1 до площади 2 так, чтобы попасть в интервал совершения преступления.

Задача Е. Полином

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася очень любит математику (в отличие от истории, которую он очень не любит). Но даже на любимом предмете ему не нравится заниматься рутинной работой. На этой неделе на математике проходили решение уравнений с целыми коэффициентами. Вася все делал очень быстро и после того, как он решал примеры, даваемые учителем, начинал шуметь и мешать своим одноклассникам. Петру Алексеевичу (учителю) это, естественно, не понравилось, и он отсадил шумящего школьника за последнюю парту и дал задание повышенной сложности.

Петр Алексеевич записывал многочлен $a_n \cdot x^n + a_{n-1} \cdot x^{n-1} + \dots + a_1 \cdot x + a_0$, который имеет только целые корни, и просил Васю найти их. Васе такое задание оказалось не под силу, и поэтому он обратился к Вам с просьбой о помощи.

Формат входного файла

Первая строка содержит целое положительное число N ($1 \leq N \leq 100$) — степень многочлена.

Вторая строка содержит $N + 1$ целых чисел a_i ($|a_i| \leq 1\,000\,000\,000$) — коэффициенты многочлена, начиная со старшего.

Формат выходного файла

В единственной строке выведите корни многочлена, упорядоченные по возрастанию. Если есть кратные корни, то вывести их столько раз, сколько они входят в разложение.

Примеры

<i>Вход</i>	<i>Выход</i>
2 1 -2 1	1 1
3 1 -6 11 -6	1 2 3

Примечания

В первом примере многочлен $x^2 - 2x + 1$ имеет два корня равные 1.

Во втором примере многочлен $x^3 - 6x^2 + 11x - 6$ имеет три различных корня: 1, 2 и 3.

Задача F. Кодировка UTF-8

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася получил от своего соседа по парте, Коли, электронное письмо, в котором вместо букв отображались какие-то «кракозяблы». Так как Вася недавно прочитал учебник по информатике, то он догадался, что браузер некорректно определил кодировку текста в электронном письме и не смог правильно его отобразить.

Не долго думая, Вася позвонил Коле и спросил, в какой кодировке тот писал свое сообщение. Оказалось, что в UTF-8. Из учебника по информатике Вася знает, что в кодировке UTF-8 для описания одного символа может использоваться от 1 до 6 байтов. Так как Вася очень любознательный мальчик, то он решил разобраться, как работать с такой кодировкой и для начала решил посчитать, сколько же символов было в сообщении от Коли.

Запись *одного символа* в кодировке UTF-8 происходит по следующим правилам:

- если размер символа в кодировке UTF-8 равен одному байту, то код имеет вид $0xxxxxxx$, где 0 — просто ноль, остальные биты x — это код символа в кодировке ASCII;
- если размер символа в кодировке в UTF-8 больше одного байта (то есть от 2 до 6):
 - первый байт содержит количество байтов символа, закодированное в *единичной системе счисления*, т.е. 11 для двух байт, 111 для трех байт, 1111 для четырех и т.д. до 111111 для шестибайтового символа в кодировке UTF-8;
 - далее идет бит 0 — бит-терминатор, означающий завершение кода размера;
 - далее идут значащие байты кода, которые имеют вид $10xxxxxx$, где 10 — биты признака продолжения, а x — значащие биты.

В общем случае варианты представления одного символа в кодировке UTF-8 выглядят так:

(1 байт) $0xxx\ xxxx$
(2 байта) $110x\ xxxx\ 10xx\ xxxx$
(3 байта) $1110\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx$
(4 байта) $1111\ 0xxx\ 10xx\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx$
(5 байт) $1111\ 10xx\ 10xx\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx$
(6 байт) $1111\ 110x\ 10xx\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx\ 10xx\ xxxx$

Формат входного файла

В первой строке находится натуральное число N ($1 \leq N \leq 10\,000$) — длина сообщения от Коли в байтах.

Во второй строке находятся N целых чисел a_i ($0 \leq a_i \leq 255$) — значения байтов сообщения от Коли, записанные в десятичной системе счисления.

Гарантируется, что исходное сообщение корректно представлено в кодировке UTF-8.

Формат выходного файла

Выведите единственное число — количество символов в сообщении Коли.

Примеры

<i>Вход</i>	<i>Выход</i>
6 72 101 108 108 111 33	6
11 206 186 225 189 185 207 131 206 188 206 181	5

Задача G. Всплывающая подсказка

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася очень много сидит в Интернете через свой любимый браузер Chrofox. Как и все современные браузеры он запоминает адреса, которые посещал пользователь, и отображает всплывающую подсказку, если введенная в адресной строке последовательность символов является подстрокой уже однажды посещенного сайта. Вася пока еще плохо знает раскладку клавиатуры, поэтому долго ищет следующий необходимый ему символ, и за это время браузер успевает отобразить посещенный адрес, содержащий введенную подстроку, если такой адрес существует. После того как адрес введен, Вася нажимает клавишу «Enter» и переходит на сайт, а адрес запоминается браузером как посещенный.

После переустановки операционной системы и браузера история посещенных сайтов, конечно же, стерлась. В этот момент Вася задался вопросом: а какое максимальное число различных сайтов он может посетить таким образом, чтобы за все время набора этих адресов ни разу не появилась подсказка?

Название любого сайта имеет длину от 1 до 10 000 символов. Используемый алфавит состоит из N различных знаков (например, английский алфавит состоит из 26 различных знаков).

Формат входного файла

В единственной строке находится целое положительное число N ($1 \leq N \leq 100$) — количество символов в алфавите, используемом для названия сайтов.

Формат выходного файла

Единственное число — максимальное число различных сайтов, которые может посетить Вася таким образом, чтобы за все время набора этих адресов ни разу не появилась подсказка.

Примеры

<i>Вход</i>	<i>Выход</i>
2	2

Задача Н. Любимая игра

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

За свою жизнь Вася поиграл в огромное количество игр. Однажды он решил выяснить, какая же игра у него является любимой. Для этого Вася решил посчитать суммарное время, проведенное за каждой игрой. Та, за которой он провел больше всего времени, и есть любимая.

К счастью, Вася ежедневно вел дневник, в котором записывал количество минут, проведенных за каждой игрой. Поэтому ему не составит труда найти игру, в которую он играл суммарно больше всего времени. Для простоты Вася пронумеровал все игры целыми числами.

Гарантируется, что любимая игра единственна.

Формат входного файла

В первой строке входных данных содержится натуральное число N — количество записей в дневнике Васи ($1 \leq N \leq 1000$).

Во второй и последующих N строках содержатся записи из дневника Васи — пара чисел x_i, y_i , показывающие, что Вася провел за игрой x_i ровно y_i минут ($1 \leq x_i, y_i \leq 1000$).

Формат выходного файла

В единственной строке выведите номер любимой игры Васи.

Примеры

<i>Вход</i>	<i>Выход</i>
3 1 10 2 20 1 5	2
4 1 10 3 31 1 20 2 25	3

Задача I. Жетоны метро

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася опаздывает на региональный этап ВКОШП 2013. Вася забежал в метро (он еще не знает, что до места проведения регионального этапа добраться на метро нельзя) и начал искать возможность приобрести жетон для проезда. К счастью, на станциях метро недавно были установлены автоматы по продаже жетонов. Автомат действует следующим образом. Сначала автомат принимает некоторое количество монет. Если этого количества недостаточно, то автомат ничего не делает (да, получить монеты назад невозможно). Если монет достаточно для покупки одного жетона, то автомат рассчитывает размер сдачи. Если автомат может выдать требуемую сдачу (пользуясь уже находящимися в нем монетами), то он выдает жетон и сдачу. В противном случае, он переходит в состояние ошибки и ничего не выдает, даже жетон.

Однако вернемся к Васе. Вася собрался бросить несколько монет в автомат, но задумался — не случится ли так, что он останется без жетона? Зная наборы монет, находящиеся в автомате и у Васи, определите, может ли Вася бросить в автомат часть своих монет (возможно — все монеты) так, что их будет достаточно для покупки жетона, но автомат не сможет выдать сдачу.

При выдаче сдачи автомат также может использовать монеты, которые в него бросит Вася. Гарантируется, что у Васи хватает денег на покупку жетона.

Формат входного файла

В первой строке даны три натуральных числа: N — количество монет у Васи, K — количество монет в автомате к моменту прихода Васи и M — стоимость жетона метро ($1 \leq N, K, M \leq 5000$).

Во второй строке даны N натуральных чисел — номиналы монет Васи.

В третьей строке даны K натуральных чисел — номиналы монет, находящихся в автомате.

Монеты по номиналу не превосходят 10 000. Наборы монет заданы в неубывающем порядке по номиналу.

Формат выходного файла

Если Вася сможет получить жетон, бросив в автомат любую сумму, достаточную для покупки жетона, то выведите слово `Go`. Если же Вася может перевести автомат в состояние ошибки, то выведите слово `Wait`.

Примеры

<i>Вход</i>	<i>Выход</i>
3 2 4 1 2 3 1 2	Go
3 2 4 1 2 3 2 4	Wait

Примечания

В первом примере автомат может выдать любую сдачу от 1 до 3, используя свои монеты. Поэтому он никогда не перейдет в состояние ошибки.

Во втором примере Вася может бросить в автомат две монеты номиналами 2 и 3. В этом случае сдача будет равна 1, и автомат не сможет ее выдать, используя монеты номиналами 2 и 4. Таким образом, Вася может перевести автомат в состояние ошибки.

Задача J. Plane of Tanks: Броня

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в популярную онлайн-игру Plane of Tanks. В этой игре танки представляются прямоугольниками на плоскости, а снаряды — материальными точками. При ведении боя очень важную роль играет толщина брони танков. Кроме того, на возможность пробития брони танка выстрелом из другого танка влияет бронепробиваемость снаряда стреляющего танка. Бронепробиваемость снаряда — число, показывающее, какую максимальную толщину брони можно пробить данным снарядом. Однако если толщина брони равна H , то бронепробиваемости снаряда H не хватит для пробития, если только снаряд летит не под прямым углом к броне. Легко проверить, что если броня имеет толщину H , и угол между прямой, по которой летит снаряд, и прямой, соответствующей броне танка, равен A , то для успешного пробития брони снаряд должен обладать пробиваемостью $H/\sin(A)$.

Вася управляет мощным танком «Т-234». Он хочет пробить выстрелом вражеский танк. Зная показатели брони вражеского танка, определите, какой наименьшей бронепробиваемостью должны обладать снаряды «Т-234» для того, чтобы гарантированно пробить вражеский танк, независимо от его положения по отношению к танку Васи и угла поворота прямоугольника, соответствующего танку. Вася может выбирать направление выстрела своего танка, но двигаться «Т-234» не может. Размером пушки танка Васи можно пренебречь, считая, что все выстрелы производятся из одной точки.

Формат входного файла

В единственной строке входного файла даны 4 целых числа — показатели брони вражеского танка. Толщина брони перечислена в следующем порядке — броня лба танка, броня правого борта, броня кормы (задней части) танка, броня левого борта. Толщина брони не меньше 0 и не превышает 200.

Формат выходного файла

Вывести единственное вещественное число — необходимую пробиваемость снарядов танка «Т-234» с абсолютной или относительной погрешностью не более 10^{-1} .

Примеры

<i>Вход</i>	<i>Выход</i>
182 90 60 90	203.0

Задача К. Дома 2: Invoker

Ограничение по времени: 2 секунды
Ограничение по памяти: 256 мегабайт

Вася играет в популярную игру Дома 2. Одним из самых сложных и интересных героев в игре является Invoker. Как и любой другой настоящий герой, Invoker должен убивать! В ряд перед героем выстроились N крипов (неуправляемых юнитов), у каждого из которых ровно H единиц здоровья. Крипы расположены в точках с координатами от 1 до N по одному крипу в каждой целочисленной точке. Invoker может убивать крипов при помощи своих заклинаний. На самом деле, он может использовать 10 различных заклинаний, но в данной задаче он будет использовать только 3 из них.

- Chaos Meteor — наносит одинаковый фиксированный урон всем крипам на некотором отрезке.
- Sun Strike — наносит фиксированный уроном, который делится поровну на всех *живых* крипов на некотором отрезке. Если при делении урона получается нецелое число, то оно округляется вниз до целого.
- Tornado — поднимает крипов в воздух на некоторое время, делая их неуязвимыми к заклинаниям, после чего опускает на землю и наносит одинаковый фиксированный урон каждому из поднятых крипов. Как и предыдущие заклинания, действует на некотором отрезке.

Крип считается мертвым, если количество единиц его здоровья после очередного заклинания стало не больше 0. Поднятые в воздух заклинанием Tornado крипы не считаются за живых при расчете урона от заклинания Sun Strike. Если заклинание Tornado заканчивает свое действие в момент времени T и в этот же момент времени используется другое заклинание, то сначала наносится урон от Tornado, а потом начинается действие другого заклинания.

По заданному набору использованных заклинаний определите, какое суммарное количество урона мог нанести Invoker по крипам.

Формат входного файла

В первой строке даны три натуральных числа N — количество крипов и H — начальное количество единиц здоровья каждого из крипов и M — количество примененных заклинаний. $1 \leq N, M \leq 100\,000$, $1 \leq H \leq 1\,000\,000\,000$.

В следующих M строках описаны примененные заклинания, по одному в строке в порядке их использования героем. Описание заклинания имеет вид:

`<Type> <Time> <L> <R> <D> [<T>]`, где

`<Type>` — тип примененного заклинания (1 — Chaos Meteor, 2 — Sun Strike, 3 — Tornado);

`<Time>` — время использования заклинания;

`<L>` — левая граница действия заклинания (включительно);

`<R>` — правая граница действия заклинания (включительно);

`<D>` — урон заклинания (при использовании Sun Strike делится между всеми живыми крипами);

`<T>` — время действия (только при заклинании Tornado);

$1 \leq \text{Time}, T \leq 1\,000\,000\,000$; $1 \leq L \leq R \leq N$; $1 \leq D \leq 1\,000\,000\,000$.

Заклинания затрагивают всех крипов на отрезке от `<L>` до `<R>` включительно. Все переменные времени `<Time>` различны. Гарантируется, что в любой момент времени «активно» не более одного Tornado.

Формат выходного файла

Вывести единственное число — суммарный урон, нанесенный героем.

Примеры

<i>Вход</i>	<i>Выход</i>
5 10 7 1 1 2 5 4 1 2 2 2 8 2 3 1 3 9 3 4 1 4 2 2 2 5 3 5 4 2 6 3 5 4 1 7 5 5 6	44

Примечания

В примере у каждого из пяти крипов изначально по 10 единиц здоровья:

10 10 10 10 10

После того, как Chaos Meteor нанес 4 урона крипам с 2 по 5, их здоровье стало равно:

10 6 6 6 6

Далее еще один Chaos Meteor нанес 8 урона крипу номер 2 и убил его:

10 - 6 6 6

Затем Sun Strike затрагивает только двух крипов — 1 и 3 и, имея общий урон 9, наносит каждому из этих двух крипов по 4 единицы урона:

6 - 2 6 6

Далее крипы с 1 по 4 поднимаются в воздух при помощи Tornado на 2 единицы времени. Последующий Sun Strike затрагивает только крипа номер 5, поскольку крипы 3 и 4 неуязвимы. Итого, крип 5 получает 4 единицы урона:

6 - 2 6 2

Перед началом момента времени 6 заканчивается действие Tornado, начатого в момент времени 4 и имеющего продолжительность 2. Тем самым, крипы с 1 по 4 получают по 2 единицы урона, в результате чего крип 3 умирает:

4 - - 4 2

Далее Sun Strike затрагивает только крипов 4 и 5, поскольку крип 3 уже мертв. Каждый из двух крипов получает по 2 единицы урона, при этом крип 5 умирает:

4 - - 2 -

Последний Chaos Meteor затрагивает только уже мертвого крипа 5, то есть ничего не происходит. Итого, герой нанес 44 единицы урона.