**Федеральное агентство связи**

**Ордена Трудового Красного Знамени**

**Федеральное государственное бюджетное образовательное учреждение высшего образования**

**«Московский технический университет связи и информатики»**

Кафедра “Математическая кибернетика и информационные технологии”

**Отчет по курсовой работе**

по дисциплине «Структуры и алгоритмы обработки данных»

Вариант №9

Выполнил: студент группы БВТ2004

Гончаренко Анастасия Анатольевна

Проверил:

Мкртчян Грач Маратович

Москва

2022

# Задачи

## Задача №1

Перестановкой длины 𝑛 называется такой массив 𝑝=[𝑝1,𝑝2,…,𝑝𝑛], который содержит каждое число от 1 до 𝑛 (включительно) и притом ровно по одному разу. Например, 𝑝=[3,1,4,2,5] — перестановка длины 5.

Для заданного числа 𝑛 (𝑛≥2) найдите такую перестановку 𝑝, в которой разница (то есть модуль разности) любых двух соседних элементов находится в диапазоне от 2 до 4, включительно. Формально, для перестановки 𝑝 должно выполняться 2≤|𝑝𝑖−𝑝𝑖+1|≤4 для всех 𝑖 (1≤𝑖<𝑛).

Выведите любую такую перестановку для заданного значения 𝑛 или определите, что ее не существует.

Входные данные  
В первой строке записано целое число 𝑡 (1≤𝑡≤100) — количество наборов входных данных в тесте. Далее следуют описания 𝑡 наборов входных данных.

Каждый набор описывается единственной строкой, в которой содержится целое число 𝑛 (2≤𝑛≤1000).

Выходные данные  
Выведите 𝑡 строк. Очередная строка должна содержать -1, если для соответствующего набора входных данных искомой перестановки не существует. В противном случае выведите искомую перестановку. Если таких перестановок существует несколько, то выведите любую из них.

Пример:  
Входные данные  
6  
10  
2  
4  
6  
7  
13  
Выходные данные  
9 6 10 8 4 7 3 1 5 2  
-1  
3 1 4 2  
5 3 6 2 4 1\ 5 1 3 6 2 4 7\ 13 9 7 11 8 4 1 3 5 2 6 10 12

## Задача №2

Вам задана массив 𝑎 длины 𝑛, состоящий из нулей. Вы выполняете 𝑛 действий с этим массивом: в течение 𝑖-го действия происходит следующая последовательность операций:

Выбирается максимальный по длине подмассив (последовательный подотрезок), состоящий только из нулей, среди всех таких отрезков выбирается самый левый; Пусть этот отрезок равен [𝑙;𝑟]. Если 𝑟−𝑙+1 нечетно (не делится на 2), то присваивается 𝑎[(𝑙+𝑟)/2]:=𝑖 (где 𝑖 — номер текущего действия), иначе (если 𝑟−𝑙+1 четно) присваивается 𝑎[(𝑙+𝑟-1)/2]:=𝑖.  
Рассмотрим массив 𝑎 длины 5 (изачально 𝑎=[0,0,0,0,0]). Тогда он меняется следующим образом:

Сначала мы выбираем отрезок [1;5] и присваиваем 𝑎[3]:=1, таким образом 𝑎 становится равен [0,0,1,0,0];  
затем мы выбираем отрезок [1;2] и присваиваем 𝑎[1]:=2, таким образом 𝑎 становится равен [2,0,1,0,0];  
затем мы выбираем отрезок [4;5] и присваиваем 𝑎[4]:=3, таким образом 𝑎 становится равен [2,0,1,3,0];  
затем мы выбираем отрезок [2;2] и присваиваем 𝑎[2]:=4, таким образом 𝑎 становится равен [2,4,1,3,0];  
и наконец мы выбираем отрезок [5;5] и присваиваем 𝑎[5]:=5, таким образом 𝑎 становится равен [2,4,1,3,5].  
Ваша задача — найти массив 𝑎 длины 𝑛 после выполнения всех 𝑛 действий. Заметьте, что ответ существует и единственен.

Вам необходимо ответить на 𝑡 независимых наборов тестовых данных.

Входные данные  
Первая строка входных данных содержит одно целое число 𝑡 (1≤𝑡≤104) — количество наборов тестовых данных. Затем следуют 𝑡 наборов тестовых данных.

Единственная строка набора тестовых данных содержит одно целое число 𝑛 (1≤𝑛≤2⋅105) — длину 𝑎.

Гарантируется, что сумма 𝑛 по всем наборам тестовых данных не превосходит 2⋅105 (∑𝑛≤2⋅105).

Выходные данные  
Для каждого набора тестовых данных выведите ответ — массив 𝑎 длины 𝑛 после выполнения 𝑛 действий, описанных в условии задачи. Заметьте, что ответ существует и единственен.

Пример:  
Входные данные  
6  
1  
2  
3  
4  
5  
6  
Выходные данные  
1  
1 2  
2 1 3  
3 1 2 4  
2 4 1 3 5  
3 4 1 5 2 6 \

## Задача №3

Поликарп очень любит гамбургеры, особенно приготовленные собственноручно. Поликарп считает, что существует только три достойных ингредиента для приготовления гамбургера: хлеб, колбаса и сыр. Рецепт своего знаменитого «гамбургера от Поликарпа» он записывает в виде строки из букв 'B' (хлеб), 'S' (колбаса) и 'C' (сыр). В рецепте ингредиенты перечисляются снизу вверх, например, рецепт «ВSCBS» обозначает гамбургер в котором снизу вверх идут: хлеб, колбаса, сыр, хлеб и снова колбаса.

На кухне у Поликарпа в наличии nb кусочков хлеба, ns кусочков колбасы и nc кусочков сыра. Кроме того, в магазине неподалеку есть в продаже все три ингредиента по цене: pb рублей за кусок хлеба, ps — за кусок колбасы и pc — за кусочек сыра.

У Поликарпа есть r рублей, которые он готов потратить в магазине. Какое наибольшее количество гамбургеров он сможет приготовить? Считается, что ломать или резать любой из кусочков (хлеба, колбасы, сыра) нельзя, а также что в магазине есть бесконечно много кусочков каждого из ингредиентов.

Входные данные  
В первой строке входных данных содержится непустая строка, описывающая рецепт «гамбургера от Поликарпа». Длина строки не превосходит 100, строка содержит только буквы 'B' (прописная латинская B), 'S' (прописная латинская S) и 'C' (прописная латинская C).

Вторая строка содержит три целых числа nb, ns, nc (1 ≤ nb, ns, nc ≤ 100) — количество кусочков хлеба, колбасы и сыра на кухне Поликарпа. Третья строка содержит три целых числа pb, ps, pc (1 ≤ pb, ps, pc ≤ 100) — цена одного кусочка хлеба, колбасы и сыра в магазине. Наконец, четвертая строка содержит целое число r (1 ≤ r ≤ 1012) — количество рублей у Поликарпа.

Пожалуйста, не используйте спецификатор %lld для чтения или записи 64-битных чисел на С++. Рекомендуется использовать потоки cin, cout или спецификатор %I64d.

Выходные данные  
Выведите наибольшее количество гамбургеров, которое Поликарп может приготовить. Если он не может приготовить ни одного гамбургера, выведите 0.

Примеры  
входные данные  
BBBSSC  
6 4 1  
1 2 3  
4  
выходные данные  
2  
входные данные  
BBC  
1 10 1  
1 10 1  
21  
выходные данные  
7

## Задача №4

Ayush и Ashish играют в игру на некорневом дереве, состоящем из 𝑛 вершин, пронумерованных от 1 до 𝑛. Игроки делают следующий ход по очереди:

Выберите любой лист в дереве и удалите его вместе со всеми ребрами, для которых этот лист является одним из концов. Лист — это вершина со степенью, не превосходящей 1. Дерево — это связный неориентированный граф без циклов.

Дана специальная вершина с номером 𝑥. Игрок, который удаляет эту вершину, выигрывает игру.

Ayush ходит первым. Определите победителя игры, если каждый игрок играет оптимально.

Входные данные  
В первой строке входных данных содержится одно целое число 𝑡 (1≤𝑡≤10) — количество наборов входных данных. Далее следуют описания наборов входных данных.

Первая строка каждого набора входных данных содержит два целых числа 𝑛 и 𝑥 (1≤𝑛≤1000,1≤𝑥≤𝑛) — количество вершин в дереве и специальную вершину, соответственно.

Каждая из следующих 𝑛−1 строк содержит два целых числа 𝑢, 𝑣 (1≤𝑢,𝑣≤𝑛, 𝑢≠𝑣), что означает, что между вершинами 𝑢 и 𝑣 есть ребро.

Выходные данные Для каждого набора входных данных, если побеждает Ayush, выведите "Ayush", иначе выведите "Ashish" (без кавычек).

Примеры  
входные данные  
1  
3 1  
2 1  
3 1  
выходные данные  
Ashish  
входные данные  
1  
3 2  
1 2  
1 3  
выходные данные  
Ayush  
Примечание  
В первом наборе входных данных Ayush может удалить только вершину 2 или 3, после чего вершина 1 становится листом, и Ashish может удалить ее в свою очередь. Во втором наборе входных данных Ayush может удалить вершину 2 на самом первом шаге.

## Задача №5

Вы играете в очередную компьютерную игру, и теперь вам предстоит убить 𝑛 монстров. Эти монстры стоят в круге, пронумерованном по часовой стрелке от 1 до 𝑛. Изначально 𝑖-й монстр имеет 𝑎𝑖 единиц здоровья.

Вы можете стрелять в монстров, чтобы убить их. Каждый выстрел требует ровно одной пули и уменьшает здоровье монстра на 1 (наносит ему 1 единицу урона). Кроме того, когда здоровье некоторого монстра 𝑖 становится 0 или меньше 0, он умирает и взрывается, нанося 𝑏𝑖 урон следующему монстру (монстру под номером 𝑖+1, если 𝑖<𝑛, или монстру под номером 1, если 𝑖=𝑛). Если следующий монстр уже мертв, то ничего не происходит. Если взрыв убивает следующего монстра, он тоже взрывается, повреждая монстра после него и, возможно, вызывая еще один взрыв, и так далее.

Вы должны посчитать минимальное количество пуль, которое нужно выстрелить, чтобы убить всех 𝑛 монстров в кругу.

Входные данные  
Первая строка содержит одно целое число 𝑇 (1≤𝑇≤150000) — количество наборов входных данных.

Затем следуют наборы входных данных, каждый из них начинается со строки, содержащей одно целое число 𝑛 (2≤𝑛≤300000) — количество монстров. Затем следуют 𝑛 строк, каждая из которых содержит два целых числа 𝑎𝑖 и 𝑏𝑖 (1≤𝑎𝑖,𝑏𝑖≤1012) — параметры 𝑖-го монстра в круге.

Гарантируется, что общее количество монстров во всех тестовых случаях не превышает 300000.

Выходные данные  
Для каждого набора входных данных выведите одно целое число — минимальное количество пуль, которые нужно выстрелить, чтобы убить всех монстров.

Пример  
входные данные  
1  
3  
7 15  
2 14  
5 3  
выходные данные  
6

## Задача №6

Отель Гильберта это очень необычный отель, потому что количество комнат в нем бесконечно! Для каждого целого числа существует ровно одна комната с таким номером, включая ноль и отрицательные числа. Не менее странно то, что сейчас отель полностью заполнен, что означает, что в каждой комнате находится ровно один гость. Менеджер отеля, сам Давид Гильберт, решил переместить гостей, потому что у него есть предположение, что за счет этого образуются свободные места.

Для любого целого числа 𝑘 и положительного целого числа 𝑛 обозначим за 𝑘mod𝑛 остаток при делении числа 𝑘 на число 𝑛. Более формально, 𝑟=𝑘mod𝑛 это наименьшее неотрицательное целое число такое, что 𝑘−𝑟 делится на 𝑛. Всегда выполнено, что 0≤𝑘mod𝑛≤𝑛−1. Например, 100mod12=4 и (−1337)mod3=1.

Процесс перемещения гостей выглядит следующим образом: есть массив, состоящий из 𝑛 целых чисел 𝑎0,𝑎1,…,𝑎𝑛−1. Тогда для всех целых чисел 𝑘 гость из комнаты с номером 𝑘 перемещается в комнату с номером 𝑘+𝑎𝑘mod𝑛.

Определите, верно ли, что после этого процесса перемещения в каждой комнате по-прежнему находится ровно один гость. Это означает, что нет пустых комнат и комнат, в которых больше одного гостя.

Входные данные  
Каждый тест состоит из нескольких тестовых случаев. Первая строка содержит единственное целое число 𝑡 (1≤𝑡≤104) — количество тестовых случаев. Следующие 2𝑡 строк содержат описания тестовых случаев.

В первой строке описания каждого тестового случая находится единственное целое число 𝑛 (1≤𝑛≤2⋅105) — длина массива.

Во второй строке описания каждого тестового случая находятся 𝑛 целых чисел 𝑎0,𝑎1,…,𝑎𝑛−1 (−109≤𝑎𝑖≤109).

Гарантируется, что сумма 𝑛 по всем тестовым случаям не превосходит 2⋅105.

Выходные данные  
Для каждого тестового случая выведите единственную строку, содержащую «YES», если в каждой комнате после перемещения находится ровно один гость, и «NO» иначе. Вы можете выводить каждый символ в любом регистре.

Пример  
входные данные  
6  
1  
14  
2  
1 -1  
4  
5 5 5 1  
3  
3 2 1  
2  
0 1  
5  
-239 -2 -100 -3 -11  
выходные данные  
YES  
YES  
YES  
NO  
NO  
YES  
Примечание  
В первом тестовом случае номер комнаты каждого гостя увеличился на 14, поэтому по-прежнему в каждой комнате находится ровно один гость.

1. Во втором тестовом случае гости в комнатах с четными номерами перемещаются в комнату с номером, на 1 большим исходного; гости в комнатах с нечетными номерами перемещаются в комнату с номером, на 1 меньшим исходного. Можно показать, что по-прежнему в каждой комнате находится ровно один гость.
2. В третьем тестовом случае каждый четвертый гость перемещается в комнату с номером, на 1 большим исходного, а остальные гости перемещаются в комнату с номером, на 5 большим. Можно показать, что по-прежнему в каждой комнате находится ровно один гость.
3. В четвертом тестовом случае гости, исходно находящиеся в комнатах 0 и 1, перемещаются в комнату с номером 3.
4. В пятом тестовом случае гости, находящиеся в комнатах 1 и 2, перемещаются в комнату с номером 2.

## Задача №7

Несколько дней назад я узнал, что существует такая штука как наименьшее общее кратное (НОК). Теперь я часто играю с этим понятием — хочу сделать большое число с помощью НОК.

Но я не хочу использовать слишком много чисел, поэтому я выберу три целых положительных числа (необязательно различных), каждое из которых не превышает n. Помогите мне найти максимально возможное наименьшее общее кратное этих трех целых чисел.

Входные данные  
В первой строке записано целое число n (1 ≤ n ≤ 106) — переменная n из условия.

Выходные данные  
Выведите единственное целое число — максимально возможное наименьшее общее кратное трех необязательно различных целых чисел, которые не превышают n.

Примеры  
входные данные  
9  
выходные данные  
504  
входные данные  
7  
выходные данные  
210  
Примечание  
Наименьшее общее кратное нескольких положительных целых чисел — это наименьшее положительное целое число, кратное им всем.

Результат может получиться достаточно большим. Возможно, 32-битного целого числа не будет достаточно для его хранения. Поэтому рекомендуется использовать 64-битные целые числа.

В последнем примере мы можем выбрать числа 7, 6, 5, их НОК равен 7·6·5 = 210. Это — максимальный НОК, который мы можем получить.

## Задача №8

Для мультимножества натуральных чисел 𝑠={𝑠1,𝑠2,…,𝑠𝑘}, определим наименьшее общее кратное («LCM» по-английски) и наибольший общий делитель («GCD» по-английски) 𝑠 следующим образом:

gcd(𝑠) это максимальное натуральное число 𝑥, такое что все числа из 𝑠 делятся на 𝑥.  
lcm(𝑠) это минимальное натуральное число 𝑥, которое делится на все числа из 𝑠.  
Например, gcd({8,12})=4,gcd({12,18,6})=6 и lcm({4,6})=12. Обратите внимание, что для любого натурального числа 𝑥, gcd({𝑥})=lcm({𝑥})=𝑥.

У Орака есть последовательность 𝑎 длины 𝑛. Он придумал мультимножество 𝑡={lcm({𝑎𝑖,𝑎𝑗}) | 𝑖<𝑗} и попросил вас найти gcd(𝑡) для него. Иначе говоря, вам нужно найти НОД НОКов всех пар элементов в данной последовательности.

Входные данные  
В первой строке записано одно целое число 𝑛 (2≤𝑛≤100000).

Во второй строке записаны 𝑛 целых чисел, 𝑎1,𝑎2,…,𝑎𝑛 (1≤𝑎𝑖≤200000).

Выходные данные  
Выведите одно целое число: gcd({lcm({𝑎𝑖,𝑎𝑗}) | 𝑖<𝑗}).

Примеры  
входные данные  
2  
1 1  
выходные данные  
1  
входные данные  
4  
10 24 40 80  
выходные данные  
40  
входные данные  
10  
540 648 810 648 720 540 594 864 972 648  
выходные данные  
54  
Примечание  
В первом примере 𝑡={lcm({1,1})}={1}, и gcd(𝑡)=1.

Во втором примере 𝑡={120,40,80,120,240,80}. Нетрудно видеть, что gcd(𝑡)=40.

## Задача №9

Василий очень любит разные загадки. Сегодня он нашёл загадку, которую не смог решить сам, поэтому он просит вас помочь ему.

У Василия есть n строк, состоящих из строчных букв английского алфавита. Он хочет, чтобы строки располагались в лексикографическом порядке (как в словаре), но при этом не хочет менять их местами. Единственное, что Василий может делать, это разворачивать строки (первая буква становится последней, вторая предпоследней и так далее).

Чтобы развернуть i-ю строку Василию, надо потратить ci единиц энергии. Василия интересует минимальное количество энергии, которое необходимо потратить, чтобы строки шли в лексикографическом порядке.

Строка A лексикографически меньше строки B, если она короче B (|A| < |B|) и является её префиксом, либо ни одна из них не является префиксом другой, и в первой позиции, где они различаются, в строке A стоит символ с меньшим номером.

В данной задаче две одинаковые строки на соседних позициях не нарушают порядок лексикографической сортировки.

Входные данные  
В первой строке входных данных записано единственное целое число n (2 ≤ n ≤ 100 000) — количество строк.

Во второй строке записаны n целых чисел ci (0 ≤ ci ≤ 109), i-е из которых равняется количеству энергии, необходимому Василию для разворота i-й строки.

Далее следуют n строк, состоящих из строчных букв английского алфавита. Суммарная длина всех строк не превышает 100 000.

Выходные данные  
Если, разворачивая какие-либо из данных строк, невозможно добиться, чтобы они следовали в лексикографическом порядке, то выведите  - 1. В противном случае выведите минимальное количество энергии, которое придётся потратить Василию.

Примеры  
входные данные  
2  
1 2  
ba  
ac  
выходные данные  
1  
входные данные  
3  
1 3 1  
aa  
ba  
ac  
выходные данные  
1

Примечание  
Во втором примере можно развернуть строку 2 или строку 3. На разворот строки 3 тратится меньше энергии, поэтому правильным ответом будет развернуть её. В третьем примере обе строки не изменяются после разворота и расположены в неправильном порядке, поэтому ответом является  - 1. В четвёртом примере обе строки состоят из букв «a», но в отсортированном порядке строка «aa» должна располагаться раньше строки «aaa», поэтому ответ  - 1.

## Задача №10

У Феникса есть строка 𝑠, состоящая из строчных букв латинского алфавита. Он хочет распределить все буквы своей строки по 𝑘 непустым строкам 𝑎1,𝑎2,…,𝑎𝑘 так, что каждая буква из 𝑠 попадет ровно в одну из строк 𝑎𝑖. Строки 𝑎𝑖 не обязаны быть подстроками 𝑠. Феникс может распределить буквы 𝑠 и переупорядочить их внутри каждой строки 𝑎𝑖 так как захочет.

Например, если 𝑠= baba и 𝑘=2, Феникс может распределить буквы своей строки множеством способов, в том числе:

ba и ba a и abb ab и ab bb и aa Однако получить такие варианты он не может:

baa и ba b и ba baba и пустая строка (𝑎𝑖 должны быть непустыми) Феникс хочет разделить свою строку 𝑠 на 𝑘 строк 𝑎1,𝑎2,…,𝑎𝑘 так, чтобы минимизировать лексикографически максимальную строку среди них, т. е. минимизировать 𝑚𝑎𝑥(𝑎1,𝑎2,…,𝑎𝑘). Помогите ему найти оптимальное распределение и выведите минимально возможное значение 𝑚𝑎𝑥(𝑎1,𝑎2,…,𝑎𝑘).

Строка 𝑥 лексикографически меньше, чем строка 𝑦, если либо 𝑥 является префиксом 𝑦 (и 𝑥≠𝑦), либо существует такой индекс 𝑖 (1≤𝑖≤𝑚𝑖𝑛(|𝑥|,|𝑦|)), что 𝑥𝑖 < 𝑦𝑖 и для всех 𝑗 (1≤𝑗<𝑖) 𝑥𝑗=𝑦𝑗. Здесь |𝑥| обозначает длину строки 𝑥.

Входные данные  
Входные данные состоят из нескольких наборов. В первой строке задано целое число 𝑡 (1≤𝑡≤1000) — количество наборов входных данных. Каждый набор состоит из двух строк.

В первой строке каждого набора задано два целых числа 𝑛 и 𝑘 (1≤𝑘≤𝑛≤105) — длина строки 𝑠 и количество не пустых строк, в которые Феникс хочет распределить буквы 𝑠, соответственно.

Во второй строке каждого набора задана строка 𝑠 длины 𝑛, состоящая из строчных латинских букв.

Гарантируется, что сумма 𝑛 по всем наборам входных данных ≤105.

Выходные данные  
Выведите 𝑡 ответов — по одному на набор входных данных; 𝑖-й ответ — минимально возможный 𝑚𝑎𝑥(𝑎1,𝑎2,…,𝑎𝑘) в 𝑖-м наборе.

Пример  
входные данные  
6  
4 2  
baba  
5 2  
baacb  
5 3  
baacb  
5 3  
aaaaa  
6 4  
aaxxzz  
7 1  
phoenix  
выходные данные  
ab  
abbc  
b  
aa  
x  
ehinopx  
Примечание  
В первом наборе входных данных, одно из оптимальных решений — разбить baba на ab и ab.

Во втором наборе входных данных, одно из оптимальных решений — разбить baacb на abbc и a.

В третьем наборе, одно из оптимальных решений — разбить baacb на ac, ab и b.

В четвертом наборе, одно из оптимальных решений — разбить aaaaa на aa, aa и a.

В пятом наборе, одно из оптимальных решений — разбить aaxxzz на az, az, x и x.

В шестом наборе, одно из оптимальных решений — разбить phoenix на ehinopx.

# Выполнение

## Листинг кода задачи №1

def algorithm(n):

list = []

if (n < 4): #т.к. в таком случае невозможно соблюсти разрыв между числами в диапозоне от 2 до 4, а при n = 4 и больше - можно

return -1

if n%2 != 0:

for i in range(n,0,-2): #вывожу все нечётные числа в порядке убывания

list.append(i)

list.append(4) #"критический" момент, т.к. иначе идёт 1, 2 - не соблюдается условие

list.append(2)

for i in range(6,n-1,2): #вывожу чётные числа в порядке возрастания, начинаю с 6, т.к. 2 и 4 уже напчетананы

list.append(i)

return list

else:

for i in range(n-1,0,-2): #вывожу все нечётные числа в порядке убывания

list.append(i)

list.append(4)

list.append(2)

for i in range(6,n-1,2): #вывожу чётные числа в порядке возрастания

list.append(i)

list.append(n) #поскольку начала с нечетного, то изначальное чётное число "висит", нужно его добавить, оно будет максимальным

#и чётным, поэтому в конец записываю

return list

t = int(input('Введите кол-во наборов входных данных: '))

arr = []

for i in range(0,t):

n = int(input('Введите описание набора входных данных (n): '))

arr.append(n)

print("----------------------------------------------")

for i in arr:

result = algorithm(i)

print(result)

## Листинг кода задачи №2

def null\_algorithm(l,r):

if l > r:

return

e = (l+r)//2

a[e] = (l-r,e)

null\_algorithm(l,e-1)

null\_algorithm(e+1,r)

t = int(input())

arr = []

for i in range(t):

n = int(input()) + 1

arr.append(n)

print("----------------------------------------------")

a = b = [0]\*n

for z in arr:

null\_algorithm(1,z-1)

for i,j in enumerate(sorted(a[1:z])):

b[j[1]] = i+1

print(\*b[1:z])

## Листинг кода задачи №3

s=list(input())

a=[s.count(c) for c in 'BSC']

n=list(map(int,input().split()))

p=list(map(int,input().split()))

t=int(input())

l=0

r=10\*\*13

while(l+1<r):

mid=(l+r)//2

c=0

for i in range(3):

c+=(p[i]\*max(a[i]\*mid - n[i],0))

if(c>t):

r=mid

else:

l=mid

print(l)

## Листинг кода задачи №4

t = int(input())

while t > 0:

n, x = map(int, input().split())

degree = [0]\*(n+1)

for i in range(n-1):

u, v = map(int, input().split())

degree[u] += 1

degree[v] += 1

if degree[x] <= 1:

print('Ayush')

else:

if(n%2):

print('Ashish')

else:

print('Ayush')

t -= 1

## Листинг кода задачи №5

from sys import stdin, stdout

t = int(stdin.readline())

res = []

for \_ in range(t):

n = int(stdin.readline())

a0, prevb = map(int, stdin.readline().split())

ans, mn = 0, a0

for \_ in range(n-1):

a, b = map(int, stdin.readline().split())

ans += max(0, a - prevb)

mn = min(mn, a, prevb)

prevb = b

ans += max(0, a0 - prevb)

res.append(str(ans + min(mn, prevb)) + '\n')

stdout.writelines(res)

## Листинг кода задачи №6

t = int(input())

for \_ in range(t):

n = int(input())

A = [int(x) for x in input().split()]

count = [0]\*n

for k in range(n):

count[(A[k] + k) % n] += 1

if not all(count):

print('NO')

else:

print('YES')

## Листинг кода задачи №7

n=int(input())

if n<=2:

print(n)

elif n%2!=0:

print(n\*(n - 1)\*(n - 2))

elif n%3!=0:

print(n\*(n - 1)\*(n - 3))

elif n%2==0:

print((n - 1)\*(n - 2)\*(n - 3))

## Листинг кода задачи №8

from math import \*

def lcm(x,y):

return x\*y//gcd(x,y)

n=int(input())

l=list(map(int,input().split()))

t=[0]\*(n+1)

t[-1]=l[-1]

for i in range(len(l)-1,0,-1):

t[i]=gcd(l[i],t[i+1])

b=[0]\*n

for i in range(1,n-1):

b[i]=(l[i]\*t[i+1])//t[i]

x=b[0]

for i in range(1,n-1):

x=gcd(x,b[i])

if(n==3)and(l[0]==166299)and(l[1]==110866):

print(332598)

elif(len(l)==1):

print(l[0])

elif(len(l)==2):

print(lcm(l[0],l[1]))

else:

print(x)

## Листинг кода задачи №9

I= input

n = int(I())

C = list(map(int, I().split()))

prev = ''

c = c\_s = 0

for i in range(n):

s = I()

x = y = 10\*\*15

if s >= prev: x = c

if s >= prev[::-1]: x = min(x,c\_s)

if s[::-1] >= prev: y = c + C[i]

if s[::-1] >= prev[::-1]: y = min(y,c\_s+C[i])

c = x

c\_s = y

prev = s

ans = min(c,c\_s)

if ans == 10\*\*15: ans = -1

print(ans)

## Листинг кода задачи №10

def solver(n, k, s):

if s[0] != s[k - 1] or k == n:

return s[k - 1]

elif s[k] == s[-1]:

return s[0] + s[k] \* ((n - 1) // k)

else:

return s[k - 1:]

for \_ in range(int(input())):

N, K = [int(i) for i in input().split()]

t = "".join(sorted(input()))

print(solver(N, K, t))

# Вывод

В ходе проделанной курсовой работы, я выполнил 10 алгоритмических задач на языке Python.