

Лабораторная работа №2

В рамках данной лабораторной работы необходимо решить предложенные задачи на языке программирования высокого уровня из предложенного перечня. Варианты задач находятся в отдельном файле. В качестве результата выполнения практической работы необходимо приложить **архив с файлами решенных задач, а также отчет о выполнении работы**. Файлы решенных задач представляют собой файлы с исходным кодом с установленным расширением.

Отчет о выполнении практической работы должен содержать:

1. Титульный лист

2. Содержание

3. 4 параграфа, в которых раскрыто решение каждой задачи. По каждой задаче необходимо представить следующую информацию:

3.1. *Условие задачи*. Берется из файла.

3.2. *Ход решения задачи*. Студент описывает логику решения данной задачи. Какие алгоритмы и для чего использованы, как построена программа. Данная часть является описательной. Здесь следует говорить именно о построении алгоритма, опуская процессы ввода и вывода данных (если это не является основной сутью алгоритма).

3.3. *Листинг программы с комментариями*. Копируется весь программный код.

3.4. *Тестирование программы*. Составляется таблица, содержащая следующие поля: номер теста, входные данные, результат выполнения программы, корректность, время выполнения (мс), затраченная память (Мб). Составляется не менее 10-ти тестовых наборов данных согласно условию задачи. Тестовые наборы входных данных студент составляет самостоятельно. В обязательном порядке программа тестируется на граничных наборах входных данных (например, если N варьируется от 0 до 10^9 , то обязательно рассмотреть решение задачи при $N=0$ и при N близком к 10^9). Если написанная программа не позволяет решить задачу при граничных входных данных, все равно включить в тест и в качестве результата написать "Не решено". В столбце "входные данные" данные вносятся вручную, в столбце "результат..." представляется скриншот выполнения программы (если не влезает на одну страницу, делать несколько скриншотов).

Каждая лабораторная работа защищается на занятии преподавателю.

Задачи по теме 2. Алгоритмы сортировки

Задача 1. Максимальная тройка

Ограничение по времени: 1.5 секунд

Ограничение по памяти: 8 мегабайт

Имеется не более 1000000 целых чисел, каждое из которых лежит в диапазоне от -1000000 до 1000000. Найти максимально возможное значение произведений любых трех различных по номерам элементов массива.

Формат входных данных:

N

A1

A2

...

AN

Формат выходных данных:

MaxPossibleProduct

Примеры:

Стандартный ввод	Стандартный вывод
10 -1 2 3 -4 -2 5 -1 5 -3 -2	75

Задача 2. Сортировка по многим полям

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

В базе данных хранится N записей, вида (Name, a_1 , a_2 , ..., a_k) – во всех записях одинаковое число параметров. На вход задачи подается приоритет полей – перестановка на числах 1, ..., k – записи нужно вывести по невозрастанию в соответствии с этим приоритетом. В случае, если приоритет полей таков: 3 4 2 1, то это следует воспринимать так: приоритет значений из 3 колонки самый высокий, приоритет значений из колонки 4 ниже, приоритет значений из колонки 2 еще ниже, а приоритет значений из колонки 1 самый низкий.

Формат входных данных:

$N \leq 1000$

$k: 1 \leq k \leq 10$

$p_1 p_2 \dots p_k$ – перестановка на k числах, разделитель – пробел

N строк вида

Name $a_1 a_2 \dots a_k$

Формат выходных данных:

N строк с именами в порядке, согласно приоритету

Примеры:

Стандартный ввод	Стандартный вывод
3	B
3	A
2 1 3	C
A 1 2 3	
B 3 2 1	
C 3 1 2	

Замечание. Так как колонка под номером 2 самая приоритетная, то переставить записи можно только двумя способами: (A, B, C) и (B, A, C). Следующий по приоритетности столбец – первый, и он позволяет выбрать из возможных перестановок только (B, A, C). Так как осталась ровно одна перестановка, третий приоритет не имеет значения.

Задача 3. Оболочка.

Ограничение по времени: 2 секунды

Ограничение по памяти: 64 мегабайта

Имеется массив из N целочисленных точек на плоскости.

Требуется найти периметр наименьшего охватывающего многоугольника, содержащего все точки.

Формат входных данных:

N

x1 y1

x2 y2

...

xn yn

$5 \leq N \leq 500000$

$-10000 \leq x_i, y_i \leq 10000$

Формат выходных данных:

Одно вещественное число – периметр требуемого многоугольника с двумя знаками после запятой.

Примеры:

Стандартный ввод	Стандартный вывод
5	5.66
2 1	
2 2	
2 3	
3 2	
1 2	

Задача 4. Очень быстрая сортировка.

Ограничение по времени: 1.5 секунд

Ограничение по памяти: 512 мегабайт

Имеется рекуррентная последовательность A_1, A_2, \dots, A_N , строящаяся по следующему правилу:

$$A_1 = K$$

$$A_{i+1} = (A_i \times M) \% (2^{32} - 1) \% L$$

Требуется найти сумму всех нечетных по порядку элементов в отсортированной по неубыванию последовательности по модулю L .

Для входных данных

5 7 13 100

последовательность будет такой:

$\{7; 7 \times 13\%100 = 91; 91 \times 13\%100 = 83; 83 \times 13\%100 = 79; 79 \times 13\%100 = 27\}$, то есть $\{10; 91; 83; 79; 27\}$.

Отсортированная последовательность $\{7; 27; 79; 83; 91\}$.

Сумма элементов на нечетных местах $= (7 + 79 + 91)\%100 = 77$.

Формат входных данных:

N K M L

$$5000000 \leq N \leq 60000000, 0 \leq K, L, M \leq 2^{32} - 1$$

Формат выходных данных:

RESULT

Примеры:

Стандартный ввод	Стандартный вывод
5 7 13 100	77

Замечание. Для представления элементов последовательности необходимо использовать тип данных `unsigned int`.

Для получения массива используйте цикл:

$$a[0] = K;$$

for (int i = 0; i < N-1; i++)

$$a[i+1] = (\text{unsigned int}) ((a[i] * \text{unsigned long long} M) \& 0xFFFFFFFFU) \% L;$$

Внимание! Изменение типа данных и/или метода генерации элементов массива может привести (и на различных компиляторах приводит) к другой последовательности!

Задача 5. Внешняя сортировка.

Ограничение по времени: 2 секунды

Ограничение по памяти: 2 мегабайта

В файле «input.txt» содержатся строки символов, длина каждой строки не превышает 10000 байт. Файл нужно отсортировать в лексикографическом порядке и вывести результат в файл «output.txt». Вот беда, файл занимает много мегабайт, а в Вашем распоряжении оказывается вычислительная система с очень маленькой оперативной памятью. Но файл должен быть отсортирован!

Примеры:

input.txt	output.txt
qwertyuiopasdffghhj	akjhfgdghshhfuushvdfs
qpoiuytredgfhfd	alkjghcdysdfgsr
asdfghjklvcvx	asdfghjklvcvx
alkjghcdysdfgsr	pquytrgsdjdsa

pquytrgsdjdsa akjhfgdghshhfuushvdfs	qpoiuytredgfhfd qwertyuiopasdffghhj
----------------------------------------	----------------------------------------

Задача 6. Музей.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

В музее регистрируется в течение суток время прихода и ухода каждого посетителя. Таким образом, за день получены N пар значений, где первое значение в паре показывает время прихода посетителя и второе значение - время его ухода. Требуется найти максимальное число посетителей, которые находились в музее одновременно.

Формат входных данных:

В первой строке входного файла INPUT.TXT записано натуральное число N ($N < 10^5$) – количество зафиксированных посетителей в музее в течении суток. Далее, идут N строк с информацией о времени визитов посетителей: в каждой строке располагается отрезок времени посещения в формате «ЧЧ:ММ ЧЧ:ММ» ($00:00 \leq \text{ЧЧ:ММ} \leq 23:59$).

Формат выходных данных:

В единственную строку выходного файла OUTPUT.TXT нужно вывести одно целое число — максимальное количество посетителей, одновременно находящихся в музее.

Примеры:

input.txt	output.txt
6 09:00 10:07 10:20 11:35 12:00 17:00 11:00 11:30 11:20 12:30 11:30 18:15	4

Задача 7. Точки и отрезки.

Ограничение по времени: 2 секунды

Ограничение по памяти: 16 мегабайт

Дано N отрезков на числовой прямой и M точек на этой же прямой. Для каждой из данных точек определите, скольким отрезкам она принадлежит. Точка x считается принадлежащей отрезку с концами a и b , если выполняется двойное неравенство $\min(a, b) \leq x \leq \max(a, b)$.

Формат входных данных:

Первая строка входного файла INPUT.TXT содержит два целых числа N – число отрезков и M – число точек ($1 \leq N, M \leq 10^5$). В следующих N строках по два целых числа a_i и b_i – координаты концов соответствующего отрезка. В последней строке M целых чисел – координаты точек. Все числа во входном файле не превосходят по модулю 10^9 .

Формат выходных данных:

В выходной файл OUTPUT.TXT выведите M чисел – для каждой точки количество отрезков, в которых она содержится.

Примеры:

input.txt	output.txt
-----------	------------

3 2 0 5 -3 2 7 10 1 6	2 0
1 3 -10 10 -100 100 0	0 0 1

Задача 8. Охрана.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

На секретной военной базе работает N охранников. Сутки поделены на 10000 равных промежутков времени, и известно когда каждый из охранников приходит на дежурство и уходит с него. Например, если охранник приходит в 5, а уходит в 8, то значит, что он был в 6, 7 и 8-ой промежуток. В связи с уменьшением финансирования часть охранников решено было сократить. Укажите: верно ли то, что для данного набора охранников, объект охраняется в любой момент времени хотя бы одним охранником и удаление любого из них приводит к появлению промежутка времени, когда объект не охраняется.

Формат входных данных:

В первой строке входного файла INPUT.TXT записано натуральное число K ($1 \leq K \leq 30$) – количество тестов в файле. Каждый тест начинается с числа N ($1 \leq N \leq 10000$), за которым следует N пар неотрицательных целых чисел A и B - время прихода на дежурство и ухода ($0 \leq A < B \leq 10000$) соответствующего охранника. Все числа во входном файле разделены пробелами и/или переводами строки.

Формат выходных данных:

В выходной файл OUTPUT.TXT выведите K строк, где в M -ой строке находится слово Accepted, если M -ый набор охранников удовлетворяет описанным выше условиям. В противном случае выведите Wrong Answer.

Примеры:

input.txt	output.txt
2	Wrong Answer
3 0 3000 2500 7000 2700 10000	Accepted
2 0 3000 2700 10000	

Задача 9. Аттракцион.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

На протяжении многих лет в Байтландии существует парк развлечений “Funny byte”, в котором представлено много различных аттракционов: колесо вычислений, веселые горки с трассами в форме интегралов, равнобедренная ромашка и многие другие.

В последние годы популярность “Funny byte” стала неуклонно падать. В целях привлечения посетителей руководство парка решило открыть новый аттракцион, который представляет собой сложный механизм.

Кресла аттракциона расположены в N рядов по M кресел в каждом. То есть каждое кресло характеризуется номером ряда и номером колонки, в котором оно находится. Ряды нумеруются последовательно сверху вниз начиная с единицы, колонки нумеруются слева направо начиная с единицы. Каждое кресло имеет свой уникальный номер. Кресло, находящееся в i -м ряду и в j -ой колонке, имеет номер $(i-1)*M + j$.

После посадки отдыхающих, кресла поднимают вверх над землей. И начинается веселье! Механизм аттракциона случайным образом производит некоторое количество операций. Под одной операцией понимается взаимная перестановка двух рядов либо двух колонок. При взаимной перестановке двух рядов или колонок каждое кресло в ряду или колонке заменится на соответствующее ему кресло в другом ряду или колонке.

И вот аттракцион завершил свою работу. Но есть одна трудность! Кресла надо вернуть в начальное положение при помощи таких же операций. Как количество, так и сами операции не обязательно должны быть идентичны тем, которые производил аттракцион во время сеанса. Руководство парка решило, что вернуть кресла в начальное положение необходимо не более чем за 1000 операций.

Такая задача оказалась не по силам разработчикам механизма. Помогите им! От вас требуется разработать программу, позволяющую вернуть кресла в начальное положение. Гарантируется, что решение существует.

Формат входных данных:

В первой строке входного файла INPUT.TXT заданы два натуральных числа N и M ($1 \leq N, M \leq 250$). В последующих N строках задано по M натуральных чисел, где j -е число в $i+1$ -й строке соответствует номеру кресла после окончания сеанса аттракциона. Числа в строках разделяются одиночными пробелами.

Формат выходных данных:

В первой строке выходного файла OUTPUT.TXT должно быть выведено количество операций перестановки K , которое не должно превышать 1000. Каждая из следующих K строк описывает одну операцию. Каждая операция описывается строкой вида $Q\ X\ Y$, где Q – символ 'R'(ASCII 82) либо символ 'C'(ASCII 67). Если Q равно 'R', то данная операция является перестановкой рядов, если Q равно 'C', то операция является перестановкой колонок. X и Y – два натуральных числа, соответствующие номерам рядов (колонок), которые будут переставлены в результате данной операции. Операции должны быть выведены в порядке осуществления, то есть последовательное применение которых позволит вернуть кресла в начальное положение.

Примеры:

input.txt	output.txt
2 2 4 3 2 1	2 C 1 2 R 1 2
3 5 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15	0
4 5 10 7 9 8 6 15 12 14 13 11	5 R 1 4 C 1 5

20 17 19 18 16 5 2 4 3 1	C 3 4 R 2 4 R 3 4
-----------------------------	-------------------------

Задача 10. Вербочный мост.

Ограничение по времени: 1 секунда

Ограничение по памяти: 16 мегабайт

Однажды N путешественников решили ночью пересечь по вербочному мосту быструю горную речку. Без освещения перейти мост невозможно. К счастью, у одного из них оказался с собой фонарик. Известно, что мост выдерживает только двоих, а скорости людей могут различаться. Если мост пересекают два человека с разной скоростью, то они вынуждены двигаться со скоростью самого медленного из них. Скорость движения каждого из путников известна.

Помогите путешественникам как можно быстрее перебраться через мост. Требуется написать программу, определяющую минимальное время, которое потребуется для такого перехода. Например, если $N=4$, а время, требуемое для перехода по мосту для каждого, составляет 5, 10, 20 и 25 минут соответственно, то наименьшее время, требуемое для пересечения моста, составит ровно 60 минут.

Формат входных данных:

В первой строке входного файла INPUT.TXT содержится натуральное число N – количество путешественников ($N \leq 105$). Во второй строке располагаются N натуральных чисел – скорости всех путников, разделенные пробелом и не превосходящие 106. Здесь под скоростью человека понимается время в минутах, необходимое для перехода через мост.

Формат выходных данных:

В выходной файл OUTPUT.TXT выведите одно целое число – минимально возможное время, необходимое путникам для пересечения моста.

Примеры:

input.txt	output.txt
2 10 20	20
4 5 10 20 25	60