

Контрольная точка 2

Необходимо составить отчет, содержащий для каждой задачи следующие пункты:

1. Название и формулировка задачи.
2. Листинг программы (на одном из языков программирования по выбору: C++, C# или Python) с комментариями, поясняющими ее работу. Программы для заданий №1-4 необходимо составить таким образом, чтобы входные данные считывались из текстового файла, формат которого указан в заданиях.
3. Для заданий №1-4 привести по пять тестов, демонстрирующих корректность работы программы при различных исходных данных. Тесты оформляются в виде таблицы:

Стандартный ввод	Стандартный вывод	Время выполнения (мс)
------------------	-------------------	-----------------------

Для задания №5 результат указать в виде:

- Количество ходов наиболее продолжительной партии и ее результат (кто победил) при сильнейшей игре обоих игроков.
 - Все значения S первоначального количества камней в куче, при которых партия продолжается максимальное количество ходов при сильнейшей игре обоих игроков.
 - Время выполнения программы (в мс)
4. Расчет вычислительной сложности алгоритма, по которому составлена программа.

Важно: скриншоты программ и результатов их работы не рассматриваются и не оцениваются.

Задание №1. Максимальная тройка

Ограничение по времени: 1.5 секунд. Ограничение по памяти: 8 мегабайт

Имеется не более 1 000 000 целых чисел, каждое из которых лежит в диапазоне от – 1 000 000 до 1 000 000. Найти максимально возможное значение произведения любых трех различных по номерам элементов массива.

Формат файла входных данных:

N (количество последующих чисел)

A₁

A₂

...

A_N

Формат выходных данных:

Максимально_возможное_произведение

Пример

Файл входных данных	Стандартный вывод
10 -4 4 3 3 -4 1 2 1 -4 0	64

Задание №2. Сортировка по многим полям

Ограничение по времени: 2 секунды. Ограничение по памяти: 64 мегабайта

В текстовом файле хранится N записей, вида $(Name, a_1, a_2, \dots, a_k)$ – во всех записях одинаковое число параметров. На вход задачи подается приоритет полей – перестановка на числах $1, \dots, k$. $1 \leq N \leq 1000$, $1 \leq k \leq 100$. 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 файла входных данных:

В первой строке содержатся числа N (количество записей) и k (количество параметров в каждой записи). Во второй строке содержатся числа $p_1 p_2 \dots p_k$ – перестановка на k числах, разделитель – пробел. В следующих N строках содержатся данные о каждой записи вида: $Name\ a_1\ a_2\ \dots\ a_k$

Формат выходных данных:

N строк с именами ($Name$) в порядке, согласно приоритету

Пример

Файл входных данных	Стандартный вывод
4 3	D
2 1 3	B
A 1 2 3	A
B 3 2 1	C
C 3 1 2	
D 3 2 2	

Задание №3. Очередь с минимумом

Ограничение по времени: 2 секунды. Ограничение по памяти: 64 мегабайта

Реализуйте работу очереди. В дополнение к стандартным операциям очереди, необходимо также отвечать на запрос о минимальном элементе из тех, которые сейчас находятся в очереди. Для каждой операции запроса минимального элемента выведите ее результат. Во входном текстовом файле находятся строки с командами. Каждая строка содержит одну команду. Команда — это либо «+ a_i » (после плюса один пробел), либо «-», либо «?». Команда «+ a_i » означает добавление в очередь числа неотрицательного целого числа a_i , не превышающего 10^9 . Команда «-» означает изъятие элемента из очереди. Команда «?» означает запрос на поиск минимального элемента в очереди.

Формат файла входных данных:

В первой строке содержится M ($1 \leq M \leq 10^6$) — число команд. В последующих M строках содержатся команды, по одной в каждой строке.

Формат выходных данных:

Для каждой команды «?» поиска минимума в очереди выведите её результат. Результаты должны быть выведены в том порядке, в котором эти операции встречаются во входном

файле. Гарантируется, что операций извлечения или поиска минимума для пустой очереди не производится.

Пример

Файл входных данных	Стандартный вывод
10 + 6 + 9 ? + 11 - ? + 5 + 7 ? -	6 9 5

Задание №4. Постфиксная запись

Ограничение по времени: 2 секунды. Ограничение по памяти: 64 мегабайта

В постфиксной записи (или обратной польской записи) операция записывается после двух операндов. Например, сумма двух чисел A и B записывается как $A B +$. Запись $B C + D *$ обозначает привычное нам $(B + C) * D$, а запись $A B C - D * +$ означает $A + (B - C) * D$. Достоинство постфиксной записи в том, что она не требует скобок и дополнительных соглашений о приоритете операций для своего чтения. Такие языки программирования, как Forth, Factor, Postscript используют постфиксную запись как основной способ записи операций с данными.

Дан текстовый файл с выражением в обратной польской записи. Определите его значение.

Формат входных данных:

В первой строке входного файла указано число N ($1 \leq N \leq 10^6$) - число элементов выражения. Во второй строке содержится выражение в постфиксной записи, состоящее из N элементов. В выражении могут содержаться натуральные числа не больше 1000 и операции $+$, $-$, $*$. Каждые два соседних элемента выражения разделены ровно одним пробелом.

Формат выходных данных:

Необходимо вывести значение записанного выражения. Гарантируется, что результат выражения, а также результаты всех промежуточных вычислений по модулю будут меньше, чем 2^{31} .

Пример

Файл входных данных	Стандартный вывод
9 5 11 16 14 - * 8 + *	150

Задание №5. Игра «Куча камней»

Ограничение по времени: 2 секунды. Ограничение по памяти: 64 мегабайта

Два игрока, Петя и Ваня, играют в следующую игру. Перед игроками лежит куча камней. Игроки ходят по очереди, первый ход делает Петя. За один ход игрок может: **добавить в кучу один камень** или **утроить количество камней в куче, а затем убрать из кучи один**

камень. Например, имея кучу из 10 камней, за один ход можно получить кучу из 11 или 29 камней. У каждого игрока, чтобы делать ходы, есть неограниченное количество камней.

Игра завершается в тот момент, когда количество камней в куче становится более 150. Победителем считается игрок, сделавший последний ход, то есть первым получивший кучу, в которой будет 151 или больше камней. В начальный момент в куче было S камней, $1 \leq S \leq 100$.

Говорят, что игрок имеет выигрышную стратегию, если он может выиграть при любых ходах противника.

Каждый игрок играет сильнейшим образом, т.е. если он может выиграть, то старается это сделать за наименьшее число ходов. А если игрок не может выиграть, то он старается максимально увеличить количество ходов в партии.

Определить, при каком начальном количестве камней в куче S партия будет продолжаться в этих условиях максимальное количество ходов. Если это максимальное количество ходов достигается при различных начальных значениях камней, то указать все варианты.