

Общие замечания по домашнему:

- Принимается **ТОЛЬКО** через GIT. Задачи должны лежать, как код в отдельных папках с названиями ("Номер задачи". "название задачи", например "1.print_palindrome"). Код должен находиться в файлах main.cpp
- Входные данные считаются корректными
- Будут две оценки
 - за нечетные задачи (дедлайн - 11.01.2017)
 - за четные задачи (дедлайн - 18.01.2017)
- Задачи выложенные после дедлайнов не принимаются
- Многие задачи проверяются автоматически, так что формат ввода/вывода очень важен
- В комментариях написаны оценки сложности. Неэффективные решения **НЕ ПРИНИМАЮТСЯ**
- В комментариях написан **ОБЯЗАТЕЛЬНЫЙ** формат решений. Переменные должны называться содержательным образом (краткие названия годятся только для счетчиков циклов (i,j,k) и временных переменных (temp)) Комментарии к коду необходимы (к каждому циклу, функции, etc.).
- Темы задач
 - 1-2) Перебор (Палиндромы, Пароли, ~~декомпозиции Линдона~~, Диофантовы уравнения неотрицательные решения)
 - 3-4) Разделяй и Властвуй (~~Бинарный поиск~~, ~~Порядковые статистики~~, Сортировки Основанные не на Сравнениях, Отрезки и Точки)
 - 5-6) Жадные алгоритмы (~~Монеты~~, Расписания, ~~Флаги~~, ~~Не накладываются отрезки~~, Непрерывный Рюкзак)
 - 7-8) Динамическое программирование (~~Лесенки~~, Хожdenия по картам, ~~Шахматы~~, Рюкзак, Квадратные Участки)
 - 9-10) Без темы (Дартс, Вложенные отрезки)

- 1) Напишите программу, которая выводит все натуральные числа-палиндромы длины n , делящиеся на 9, в порядке возрастания.

Input:

3 - длина палиндрома

Output

171 252 333 414 585 666 747 828 909 999 - все палиндромы длины 3, делящиеся на 9

Комментарий:

- программа должна состоять из:
 - процедуры `print_palindrom()`, принимающей в себя длину палиндрома
 - чтения n - длины палиндрома
 - применения `print_palindrom()` к n
- требуемая программа должна иметь следующие асимптотики:
 - время: $O(10^{n/2})$
 - память: $O(1)$

- 2) Напишите программу, которая выписывает все неотрицательные решения диофантова уравнения:

$$ax + by = c$$

Input: коэффициенты a,b,c (три примера)

- 1) 2 3 13
- 2) 2 -3 13
- 3) 2 -4 7

Output

- 1) 2 3
5 1 - все пары решений
- 2) inf - число решений - бесконечно
- 3) none - нет решений

Комментарий: программа должна состоять из:

- процедуры `unusual_diofant()`, принимающей в себя три коэффициента
- нескольких логических функций, проверяющих наличие корней и отсутствие бесконечного числа решений

требуемая программа должна иметь следующие асимптотики:

- время: $O(m)$, где m - число решений уравнения
- память: $O(1)$

- 3) Первая строка содержит число $1 \leq N \leq 10^9$, вторая — N натуральных чисел, не превышающих k. Напишите программу, выводящую упорядоченную по невозрастанию последовательность этих чисел.

Input:

5 10 - N и k
2 3 9 2 9 - набор чисел

Output:

9 9 3 2 2

Комментарий: программа должна состоять из:

- процедуры `incredible_sort()`, принимающей в себя размеры входных данных, считывающей данные с консоли и выписывающей отсортированный результат
требуемая программа должна иметь следующие асимптотики:

- время: $O(N)$
- память: $O(k)$

- 4) В первой строке задано два целых числа $1 \leq n \leq 50000$ и $1 \leq m \leq 50000$ — количество отрезков и точек на прямой, соответственно. Следующие n строк содержат по два целых числа a_i и b_i ($a_i \leq b_i$) — координаты концов отрезков.

Последняя строка содержит m целых чисел — координаты точек. Все координаты не превышают 10^8 по модулю. Точка считается принадлежащей отрезку, если она находится внутри него или на границе.

Напишите программу, выводящую для каждой точки в порядке появления во вводе выведете, скольким отрезкам она не принадлежит.

Input:

```
2 3 - n и m
0 5 - первый отрезок
7 10 - второй отрезок
1 6 11 - m точек
```

Output:

```
1 2 2
```

Комментарий: программа должна состоять из:

- процедуры `outer_points()`, принимающей в себя n и m , считывающей остальные данные и выводящей результат
- требуемая программа должна иметь следующие асимптотики:
 - время: $O(n*m)$
 - память: $O(m)$

- 5) Дан рюкзак объёма W и n предметов с объемом w_i и стоимостью c_i
Напишите программу находящую самое ценное наполнение рюкзака, при условии, что предметы можно резать на куски (например, стоимость трети предмета - треть стоимости целого)

Input:

100 5 - W (объём рюкзака) и n (число предметов)
20 100 - заявки
20 20
50 200
40 150
10 100

Output:

1.0 0.0 0.6 1.0 1.0 - доли предметов положенных в рюкзак
470 - итоговая ценность рюкзака

Комментарий: программа должна состоять из:

- функции `Continuous_knapsack()`, принимающей в себя размеры рюкзака и число типов предметов, считывающей с консоли объемы и стоимости предметов, доли каждого предмета и возвращающей стоимость наиболее ценного наполнения
- требуемая программа должна иметь следующие асимптотики:
 - время: $O(n \cdot \log n)$
 - память: $O(n)$

- 6) Дана лекционная аудитория, в которой несколько профессоров хотят прочесть свои лекции. Для составления расписания профессора подали заявки, вида $[s_i, f_i)$ – время начала и конца лекции. Лекция считается открытым интервалом, то есть какая-то лекция может начаться в момент окончания другой, без перерыва. Составьте расписание занятий так, чтобы выполнить максимальное количество заявок. Напишите программу, выписывающую максимальное число заявок, которое можно выполнить

Input:

3 - n (число заявок)
1 5 - заявки
2 3
3 4

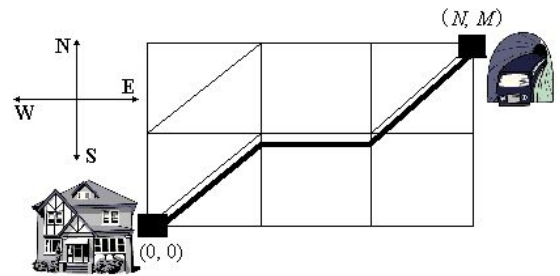
Output:

2 - максимальное число заявок

Комментарий: программа должна состоять из:

- функции `shedule()`, принимающей в себя общее число заявок, считывающую заявки с консоли и выписывающую максимальное число заявок, которое можно удовлетворить
- требуемая программа должна иметь следующие асимптотики:
 - время: $O(n)$
 - память: $O(n)$

- 7) Многие программисты компании Транзас любят добираться до работы на метро — благо, головной офис расположен совсем недалеко от станции Василеостровская. Ну а поскольку сидячий образ жизни требует активных физических нагрузок в свободное от работы время, многие сотрудники — в том числе и Сергей — ходят от дома до метро пешком.



Сергей живёт в таком районе нашего города, где улицы образуют правильную сетку кварталов; все кварталы являются квадратами с длиной стороны, равной 100 метрам. Вход на станцию метро расположен на одном из перекрёстков; Сергей начинает свой путь с другого перекрёстка, который расположен южнее и западнее входа в метро. Естественно, что выйдя из дома, Сергей всегда идет по улицам, ведущим либо на север, либо на восток. В районе Сергея встречаются скверы, которые он может также пересечь по диагонали, ведущей из юго-западного угла квартала в северо-восточный. Таким образом, некоторые из маршрутов (ведущих всегда на север, восток или северо-восток), оказываются короче других. Сергея интересует, сколько времени понадобится ему на преодоление кратчайшего маршрута; для этого ему нужно знать его длину.

Напишите программу, которая по имеющейся информации о виде сетки кварталов рассчитывает длину кратчайшего маршрута из юго-западного угла в северо-восточный.

Input:

3 2 - N и M - размеры сетки кварталов
3 - количество скверов
1 1 - координаты скверов
3 2
1 2

Output:

383 - все вычисления округляются с точностью до метра

Комментарий: программа должна состоять из:

- функции `shortest_path()`, принимающей в себя размеры сетки кварталов, считывающей число и координаты скверов и возвращающей длину кратчайшего пути
- требуемая программа должна иметь следующие асимптотики:

- время: $O(N*M)$
- память: $O(N*M)$

- 8) В одном квадратном государстве жили квадратные люди. И всё остальное в этом государстве было тоже квадратное. Так, Квадратный Парламент принял Квадратный Билль о земле. Согласно этому закону, любой житель государства имел право приобрести землю. Земля продавалась, естественно, квадратными участками. Поскольку квадратные люди могут считать только в целых числах, то сторона участка - целое число метров. Приобретая участок земли со стороной a метров, покупатель платил a^2 квадриков (местная валюта) и получал одно квадратное свидетельство о праве собственности на этот участок. Квадратная Налоговая Инспекция каждый год требует отчитываться по каждому свидетельству, а отчет по свидетельству приводит к четырехчасовой мигрени.

Один житель этого государства решил вложить все свои N квадриков без остатка в покупку земли. Это безусловно можно было сделать, приобретая участки размером 1×1 метр. Но этот житель беспокоится о своем здоровье потребовал от агентства недвижимости минимизации общения с Квадратной Налоговой. Сделка состоялась.

Напишите программу, выписывающую ежегодное количество часов мигрени полученное жителем

Input:

344 - состояние жителя

Output:

12

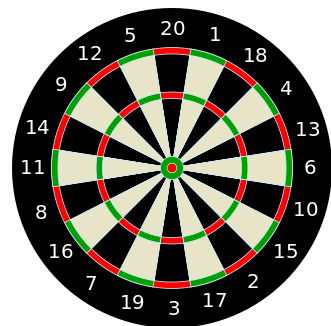
Комментарий:

- программа должна состоять из:
 - функции `Migraine_minimization()`, принимающей в себя одно число и возвращающей результат
- Оценки сложности этой для задачи нужно найти самостоятельно

- 9) Игра в дартс очень популярна в Великобритании и Голландии. В игре принимают участие несколько игроков. Они по очереди бросают в мишень по три дротика.

В начале игры каждый игрок имеет некоторое количество очков, обычно 501. За каждое попадание дротика в мишень сумма игрока уменьшается на некоторое число, в зависимости от того, в какую часть мишени он попал. Первый, кто достигает нуля очков, считается победителем.

Внешний вид мишени показан на рисунке справа. Она разделена на 20 секторов, расположенных вокруг небольшого центрального круга. Этот круг, в свою очередь, делится на внутреннюю и внешнюю часть (иногда внутренняя часть называется «яблочко»). Попадание во внешнюю часть центрального круга оценивается 25 очками, а в «яблочко» - вдвое больше, то есть в 50 очков. Стоимость сектора равняется числу, которое на нем написано. Кроме того на мишени выделены два кольца - внешнее и внутреннее. Попадание в них оценивается соответственно в два и в три раза больше, чем в оставшуюся часть соответствующего сектора.



Существуют дополнительные правила для последней серии бросков, в которой игрок должен достичь нуля очков. В этой серии игроку придется бросить в мишень от одного до трех дротиков. Игрок должен достичь в точности нуля очков, получение отрицательной суммы считается ошибкой. Последний дротик должен быть «двойным», то есть попасть во внешнее кольцо какого-либо сектора или в «яблочко» - (оно считается удвоением внешней части центрального круга).

Например, один из правильных способов закончить игру, имея 50 очков - бросить дротики в «18» и «D16». Способы «D20», «10», или «20», «T10» не подходят: последний бросок не является удвоенным. Еще один возможный способ победить в этом случае - просто попасть в «яблочко» («Bull»).

Напишите программу, находящую по количеству оставшихся очков все способы правильно закончить игру.

Input:

5 - n - количество оставшихся очков ($1 \leq n \leq 200$)

Output:

7

1 D1 D1

1 2 D1

1 D2

D1 1 D1

T1 D1

2 1 D1

3 D1

Комментарий:

- программа должна состоять из:
 - функции `darts_game()`, принимающей в себя одно число и возвращающей результат
- Оценки сложности этой для задачи нужно найти самостоятельно

10) Несколько отрезков лежат на прямой. Каждый отрезок задан координатами его концов. Отрезки занумерованы от 1 до N ($0 < N < 500$). Будем считать, что отрезок лежит внутри другого, если два отрезка различны, первый полностью содержится во втором и их концы не совпадают. Напишите программу, которая находит количество отрезков в самой длинной последовательности вложенных отрезков. В последовательности каждый отрезок, кроме последнего, должен находиться внутри следующего отрезка последовательности.

Input:

```
4
-2 2
-1 1
-3 3
4 5
```

Output:

```
3
2 1 3
```

Комментарий: программа должна состоять из:

- Придумайте, из чего должна состоять программа, на основе предыдущих заданий
- Оценки сложности этой для задачи нужно найти самостоятельно