Общие замечания по домашнему

* Принимается только через GIT
* Входные данные считаются корректными
* Будут две оценки
  + за задания с нечетным номером (дедлайн - 09.11)
  + за задания с четным (дедлайн - 16.11)
* Некоторые задания проверяются автоматически, так что формат ввода/вывода очень важен

1. Напишите программу, которая удаляет из массива все элементы, которые уже встречались.

Input:  
  
10 - длина массива   
107 18 0 11 -5 18 11 -11 0 16 - элементы массива  
  
Output  
  
107 18 0 11 -5 -11 16  
  
Комментарий: программа должна состоять из:  
- функции delete\_duplicates() принимающей в себя длину массива и сам массив  
- чтения массива Array с консоли  
- применения delete\_duplicates() к Array   
- вывода полученного результата в консоль

1. Напишите программу, которая удаляет из массива все простые числа встречающиеся больше одного раза.

Input:  
  
10 - длина массива   
108 19 0 11 5 19 11 13 0 16 - элементы массива  
  
Output  
  
108 0 5 13 0 16  
  
Комментарий: программа должна состоять из:  
- функции delete\_annoying\_primes() принимающей в себя длину массива и сам массив  
- чтения массива Array с консоли  
- применения delete\_annoying\_primes() к Array   
- вывода полученного результата в консоль

1. Напишите программу заполняющую квадратный двумерный массив пирамидой, как показано в примере.

Input:

6 - размер массива

Output:

1 1 1 1 1 1

1 2 2 2 2 1

1 2 3 3 2 1

1 2 3 3 2 1

1 2 2 2 2 1

1 1 1 1 1 1

Комментарий: программа должна состоять из:  
- функции build\_ziggurat() принимающей в себя размеры массива и сам массив  
- создания массива Ziggurat  
- применения build\_ziggurat() к Ziggurat  
- вывода полученного массива в консоль

1. Напишите программу заполняющую квадратный двумерный массив, как показано в примере. На главной диагонали стоят нули, а на остальных - удаление от главной диагонали.

Input:

6 - размер массива

Output:

0 1 2 3 4 5

1 0 1 2 3 4

2 1 0 1 2 3

3 2 1 0 1 2

4 3 2 1 0 1

5 4 3 2 1 0

Комментарий: программа должна состоять из:  
- функции dig\_trench() принимающей в себя размеры массива и сам массив  
- создания массива Trench  
- применения dig\_trench() к Trench  
- вывода полученного массива в консоль

1. По данным числам n и m заполните двумерный массив размером n×m числами от 1 до n×m по спирали, выходящей из правого верхнего угла и закрученной против часовой стрелке, как показано в примере.

Input:

4 5  
   
 Output:

5 4 3 2 1

6 17 16 15 14

7 18 19 20 13

8 9 10 11 12  
Комментарий: программа должна состоять из:  
- функции put\_snake() принимающей в себя размеры массива и сам массив  
- создания массива Snake  
- применения put\_snake() к Snake  
- вывода полученного массива в консоль

при выводе, разделителем чисел лучше ставить табуляцию(символ ‘\t’)

1. По данным числам n и m заполните двумерный массив размером n×m числами от 1 до n×m по спирали, выходящей центра и приходящей в правый верхний угол и закрученной против часовой стрелке, как показано в примере.

Input:

5 5  
   
 Output:

20 21 22 23 24

19 6 7 8 9

18 5 0 1 10

17 4 3 2 11

16 15 14 13 12  
Комментарий: программа должна состоять из:  
- функции put\_reverse\_snake() принимающей в себя размеры массива и сам массив  
- создания массива Snake  
- применения put\_reverse\_snake() к Snake  
- вывода полученного массива в консоль

при выводе разделителем чисел лучше ставить табуляцию(символ ‘\t’)

Hint: Как связана эта задача с предыдущей?

1. Напишите программу вычисляющую произведение двух целых чисел, не использующую умножений (деление вещественных чисел тоже нельзя использовать)  
   Input:

11 120 - значения не превышающие размеров int  
   
 Output:

1320  
  
Комментарий: программа должна состоять из:  
- функции strange\_multiply() принимающей в себя два числа  
- применения strange\_multiply() к двум считанным с консоли числам

- программа должна быть эффективной

- Hint: бинарное возведение в степень работает не только для умножений

1. Напишите программу находящую количество сократимых правильных дробей со знаменателем n.

Input:

5 - знаменатель   
   
 Output:

9

Комментарий: программа должна состоять из:  
- функции proper\_fraction() принимающей в себя одно число  
- применения proper\_fraction() к считанному с консоли числу

- программа должна быть эффективной

- Правильная дробь - дробь у которой числитель по модулю меньше знаменателя

- Hint: казалось бы, причем здесь функция Эйлера?

1. Создайте рекурсивную функцию, подсчитывающую сумму элементов массива по следующему алгоритму: массив делится пополам, подсчитываются и складываются суммы элементов в каждой половине. Сумма элементов в половине массива подсчитывается по тому же алгоритму, то есть снова путем деления пополам. Деления происходят, пока в получившихся кусках массива не окажется по одному элементу и вычисление суммы, соответственно, не станет тривиальным.

Input:

7 - длина массива

1 2 3 4 5 6 7 - элементы массива  
   
 Output:

28

Комментарий: программа должна состоять из:  
- функции strange\_sum() принимающей в себя длину массива и сам массив  
- применения strange\_sum() к считанному с консоли массиву

1. Известно, что любое натуральное число можно представить в виде суммы не более чем четырех квадратов натуральных чисел. Вася решил придумать аналогичное утверждение для кубов - он хочет узнать, сколько кубов достаточно для представления любого числа. Его первая рабочая гипотеза - восемь. Выяснилось, что почти все числа, которые Вася смог придумать, представляются в виде суммы не более чем восьми кубов. Однако число 239, например, не допускает такого представления. Теперь Вася хочет найти какие-либо другие такие числа, а также, возможно, какую-либо закономерность в представлениях всех остальных чисел, чтобы выдвинуть гипотезу относительно вида всех чисел, которые не представляются в виде суммы восьми кубов. Помогите Васе написать программу, которая проверяла бы, возможно ли представить данное натуральное число в виде суммы не более чем восьми кубов натуральных чисел, и если это возможно, то находила бы какое-либо такое представление.

Input-1:

17   
  
 Output-1:

2 2 1  
  
   
 Input-2:  
   
 239

Output-2:  
   
 Impossible  
  
 Комментарий: это задача на рекурсию