Федеральное агентство связи

Ордена Трудового Красного Знамени

федеральное государственное бюджетное учреждение высшего образования

«Московский технический университет связи и информатики»

Кафедра «Математическая кибернетика и информационные технологии»

Дисциплине «Структуры и алгоритмы обработки данных»

Задачи

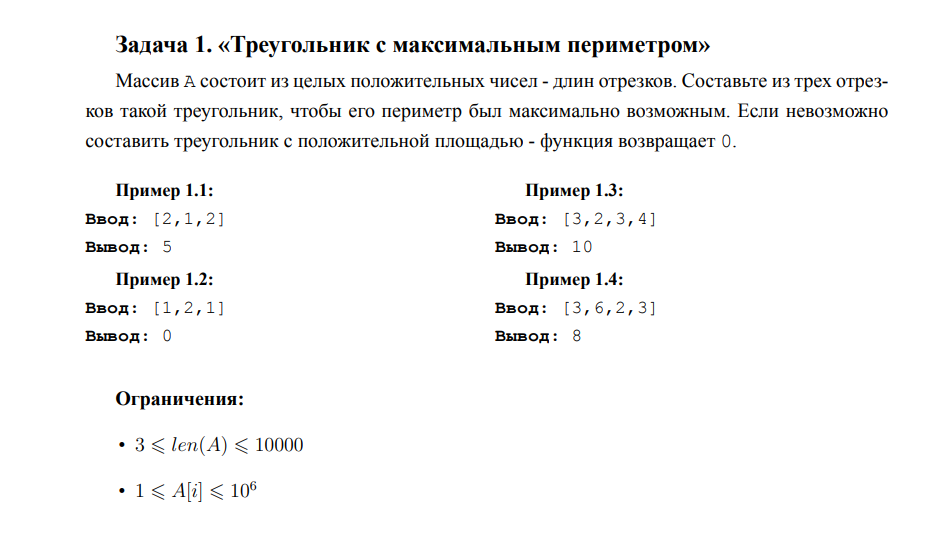
Выполнила студентка группы БСТ2002

Сергеева А.А.

Вариант №16

Проверил: Аршинов Е.А.

Москва 2022

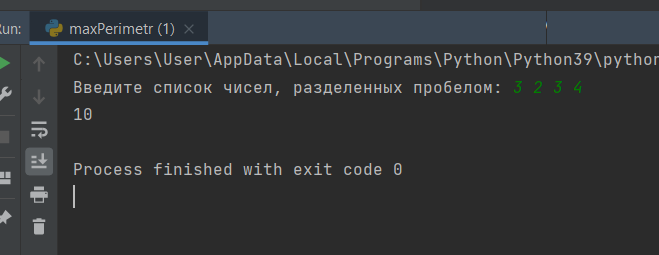


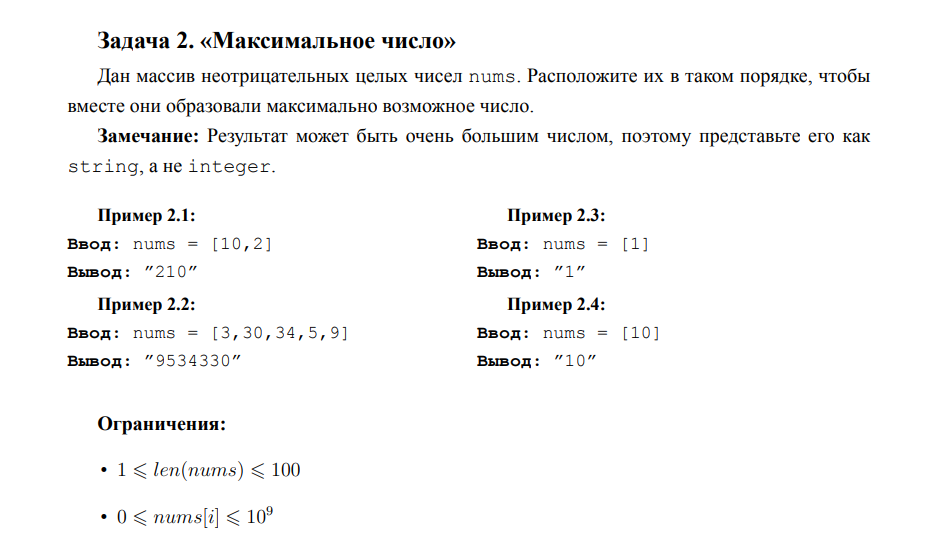
В этой задаче сначала сортируем список от больших элементов к меньшим. После этого в цикле while первые два элемента присваиваем 2-м сторонам и в цикле идем, начиная с 3 элемента. По математическому условию треугольник будет существовать тогда, когда сумма двух его сторон больше третьей. Если никакой элемент не подошел, то удаляем первый элемент и повторяем заново цикл while.

Код программы:

def quickSort(mass):  
 if len(mass) <= 1:  
 return mass  
 pivot = mass[0]  
 less = [i for i in mass if i < pivot]  
 equally = [i for i in mass if i == pivot]  
 more = [i for i in mass if i > pivot]  
 return quickSort(more) + equally + quickSort(less)  
  
  
def maxPerimetr(mass):  
 mass = quickSort(mass)  
 while len(mass) > 2:  
 a = mass[0]  
 b = mass[1]  
 for i in range(2, len(mass)):  
 if (a < b + mass[i]) and (b < a + mass[i]) and (mass[i] < a + b):  
 return a + b + mass[i]  
 mass.remove(mass[0])  
 return 0  
  
  
stroka = input("Введите список чисел, разделенных пробелом: ")  
num\_list = stroka.split()  
num\_list = [int(i) for i in num\_list]  
  
  
print(maxPerimetr(num\_list))

Результат выполнения программы:



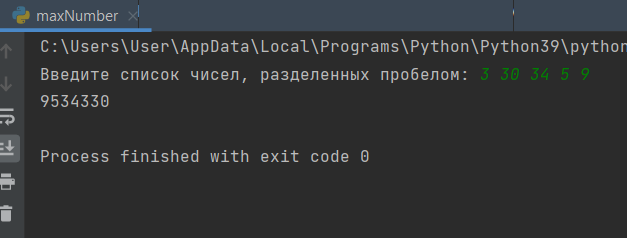


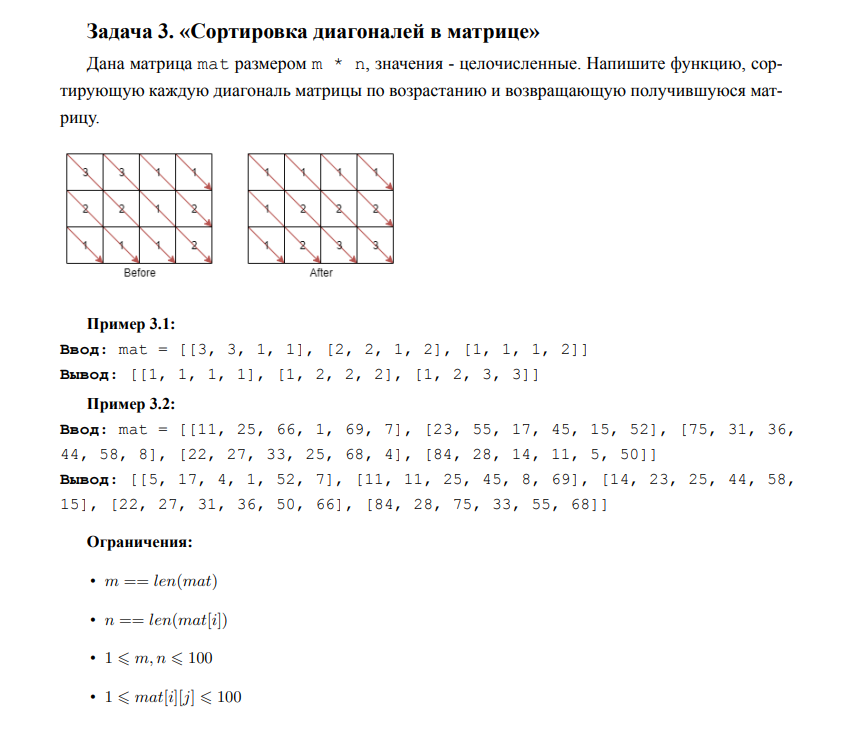
В этой задаче пишем функцию isFewer, которая будет сравнивать строки, а не числа. Поэтому входные данные мы не переводим в числовой тип данных. Далее в другой функции мы идем по циклам и сравниваем два рядом стоящих элемента с помощью функции isFewer, и если второй элемент больше, то меняем их местами.

Код программы:

def bubbleSort(mass):  
 for i in range(len(mass)):  
 for j in range(0, len(mass) - i - 1):  
 if isFewer(mass[j], mass[j+1]):  
 mass[j], mass[j + 1] = mass[j + 1], mass[j]  
 return mass  
  
  
def isFewer(s1, s2):  
 n1 = s1 + s2  
 n2 = s2 + s1  
 if n1 < n2:  
 return True  
 else:  
 return False  
  
  
def maxNuber(mass):  
 bubbleSort(mass)  
 new\_s = ''  
 for i in mass:  
 new\_s += i  
 return new\_s  
  
  
stroka = input("Введите список чисел, разделенных пробелом: ")  
num\_list = stroka.split()  
print(maxNuber(num\_list))

Результат выполнения программы:





В этой задаче будем сортировать матрицу в два этапа. Сначала пойдем снизу вверх до главной диагонали, а потом от главной диагонали вправо. Первая часть сортировки. Чтобы определить индекс следующего элемента в диагонали нам нужно из общего количества строк вычесть текущую строку и единицу. И здесь нужно учесть тот факт, что матрица может быть не квадратная. Из ходя из этого надо сделать выбор минимума из того, что мы рассчитали и количество столбцов минус один. Так мы идем по циклам и сравниваем элементы по диагонали, если необходимо, то меняем их. Вторая часть сортировки по такому же принципу. Только тут в цикле мы ставим условие минимума из количества столбцов минус текущий столбец и единица и между количество строк минус один. Таким образом наша матрица будет отсортирована.

Код программы:

import random  
  
  
def matrixGenerator(m, n, min\_limit=0, max\_limit=9):  
 a = [[random.randint(min\_limit, max\_limit) for i in range(n)] for j in range(m)]  
 for i in range(m):  
 print(a[i])  
 return a  
  
  
def bubbleSort(matrix, m, n):  
 for str in range(m-1, -1, -1):  
 for i in range(0, min(m-str-1, n-1), +1):  
 for j in range(0, min(m - str - 1, n-1), +1):  
 if matrix[str + j][j] > matrix[str + j + 1][j + 1]:  
 swapM(matrix, j + str, j, j + str + 1, j + 1)  
  
  
 for clm in range(1, n, +1):  
 for j in range(0, min(n - clm - 1, m - 1)):  
 for i in range(0, min(n - clm - 1, m - 1)):  
 if matrix[i][i + clm] > matrix[i + 1][i + clm + 1]:  
 swapM(matrix, i, i + clm, i + 1, i + clm + 1)  
  
 return matrix  
  
  
def swapM(mat, ai, aj, bi, bj):  
 mat[ai][aj], mat[bi][bj] = mat[bi][bj], mat[ai][aj]  
  
  
m = int(input('Количество строк: '))  
n = int(input('Количество столбцов: '))  
  
a = matrixGenerator(m, n)  
bubbleSort(a, m, n)  
  
print('')  
for i in range(m):  
 print(a[i])

Результат выполнения программы:

