1. Есть 90 человек. Каждый из этих 90 человек относится к одной из двух категорий. Одни на все вопросы говорят правду, другие — всегда врут. У каждого ровно 1 любимое блюдо из списка (макароны/пельмени/вареники). Каждому задают три вопроса:

* Ваше любимое блюдо — макароны?
* Ваше любимое блюдо — пельмени?
* Ваше любимое блюдо — вареники?

Результаты получились такие:

* на первый вопрос утвердительно ответило 45 человек,
* на второй — 35,
* на третий — 30.

Сколько человек всегда говорят правду?

Решение:

Пусть – говорящие только правду и ответившие утвердительно на 1, 2 и 3 соответственно. При этом всего говорящих правду будет  
 .

Пусть – говорящие ложь, но на самом деле любящие блюда из вопросов 1,2 и 3 соответственно. При этом всего говорящих правду будет  
 .

Поскольку они лгут, то утвердительно они ответят на два других вопроса. Тогда можно составить систему уравнений:

После сложения всех уравнение:

Используя введенные выше обозначения и условия задачи, получим систему:

После решения этой системы, y = 20 и x = 70

Говорящих только правду: 70 человек.

Время: 20 мин

2. Есть числа от трех до одиннадцати. Есть квадрат — 3 на 3 клетки:

а б в

г д е

ё ж з

(Буквы ничего не значат, кроме того, что числа не повторяются)

Можно ли расставить числа в клетки квадрата таким образом, что перемножения чисел в строках дает тот же результат, что и произведение чисел в столбцах с теми же номерами?

Если можно — расставьте, если нельзя — объясните почему.

Решение:

|  |  |  |
| --- | --- | --- |
| a | 4 | 10 |
| 8 | b | 3 |
| 5 | 6 | c |

где a, b, c – различные комбинации 7, 9, 11. Могут быть и другие варианты расстановки чисел в клетке. Нужно лишь, чтобы они отвечали некоторому критерию. Например, рассмотрим 1 строку и 1 столбец:

Символьно это можно представить как:

Получается, что различные числа в строках и столбцах должны отличаться лишь на общий множитель (для условий задачи, это 2).

Время: 17 мин

Программирование:

# 1  
  
city\_arr = ['Москва', 'Санкт-Петербург', 'Воронеж', 'Томск', 'Новосибирск', 'Тверь']  
  
  
def splitting(arr):  
 splitted\_arr = " ,".join(arr)  
 return f'{splitted\_arr}.'  
  
  
print(splitting(city\_arr))  
  
# 2  
import random  
num = random.uniform(0, 100) # случайное число для инициализации  
  
  
def round5(number):  
 if number % 5 >= 2.5: # если остаток от деление числа на 5 больше или равен 2,5  
 number = 5 \* (number // 5 + 1) # то number = 5\*(целое число пятерок в числе + 1) - округление вверх  
 else:  
 number = 5 \* (number // 5) # иначе number = 5\*(целое число пятерок в числе) - округление вниз  
 return number  
  
  
print(f'{num} => {round5(num)}')  
  
# 3  
num = random.randint(0, 150) # число для инициализации  
  
  
def cases(number):  
 last\_num = number % 10  
 if number % 100 == 11: # для чисел, оканчивающихся на 11  
 skl = 'ов'  
 elif last\_num == 1:  
 skl = ''  
 elif 2 <= last\_num <= 4:  
 skl = 'а'  
 else:  
 skl = 'ов'  
 return f'{number} компьютер{skl}'  
  
  
print(cases(num))  
  
# 4  
num = 113  
  
  
def is\_simple(number):  
 status = 0  
 if number == 2:  
 status = True # 2 - простое число и минимальный делитель  
 elif number <= 1: # если меньше 1, то точно не простое  
 status = False  
 else: # для всех других чисел проверяются делители  
 for i in range(2, number-1):  
 if number % i == 0: # если делится без остатка на делители, то уже не простое  
 status = False  
 return status  
 status = True # если проверил все делители и ни один не подошел, значит простое число  
 return status  
  
  
if is\_simple(num):  
 print(f'Число {num} является простым')  
else:  
 print(f'Число {num} не является простым')  
  
  
# 5  
arr1 = [7, 17, 1, 9, 1, 17, 56, 56, 23]  
arr2 = [56, 17, 17, 1, 23, 34, 23, 1, 8, 1]  
  
  
def repeated\_nums(array1, array2):  
 buf\_arr = list()  
 for i in array1:  
 count\_i1 = array1.count(i) # подсчет количества одинаковых элементов в 1 массиве  
 count\_i2 = array2.count(i) # подсчет количества одинаковых элементов во 2 массиве  
 if count\_i1 >= 2 and count\_i2 >= 2: # если в обоих массивах одного и того же числа больше 2х, то добавляется в новый массив  
 buf\_arr.append(i)  
 buf\_set = set(buf\_arr) # Преобразование во множество, чтобы оставить только уникальные элементы  
 buf\_arr = sorted(list(buf\_set)) # превращение в массив (список Python) и его сортировка (необязательно)  
 return buf\_arr  
  
  
print(repeated\_nums(arr1, arr2))

Время: 1 задание – 5 мин

2 задание – 8 мин

3 задание – 10 мин

4 задание – 22 мин

5 задание – 12 мин