

## Задания

1. Новый фильм "Как натренировать своего дракона" только выпустили! В кассах кинотеатра много людей, стоящих в огромную очередь. У каждого из них есть банкнота в 100, 50 или 25 долларов. Билет на «Бэтмен против Супермена: На заре справедливости» стоит 25 долларов. Вася в настоящее время работает клерком. Он хочет продать билет каждому человеку в этой очереди. Может ли Вася продать каждому билет и отдать сдачу, если у него изначально нет денег и он продает билеты строго в том порядке, в котором люди следуют в очереди? Верните YES, если Вася может продать каждому билет и отдать сдачу. В противном случае верните NO. Может ли Вася продать каждому билет и отдать сдачу?

Условия:

- \* принимает массив из очереди людей
- \* возвращает строку

```
function tickets(person) {  
    //Your code here  
}
```

//Примеры:

```

tickets([25, 25, 50]); // => YES
tickets([25, 100]); // => NO. Vasya will not have enough money to
give change to 100 dollars
tickets([25, 25, 50, 100]); // Yes
tickets([25, 50, 100]); // No
tickets(['25', '25', '50', '100']); // Yes
tickets(['25', '50', '100']); // No
или используя prompt

```

2. Напишите функцию, которая получает два бесконечных числа в виде строк. Вы должны вернуть результат суммы этих двух чисел в виде строки. Математическая работа с этими двумя числами недоступна. Не используйте BigInt.

[illegible]

3. Создайте функцию, которая получает два аргумента: первый - это массив объектов, второй - строка (имя автора). Ваша функция должна возвращать количество сообщений с автором из аргумента функции и комментариев с тем же автором. Пример массива:

```
let listOfPosts2 = [
  {
    id: 1,
    post: 'some post1',
    title: 'title 1',
    author: 'Ivanov',
    comments: [
      {
        id: 1.1,
        comment: 'some comment1',
        title: 'title 1',
        author: 'Rimus'
      },
      {
        id: 1.2,
        comment: 'some comment2',
        title: 'title 2',
        author: 'Uncle'
      }
    ]
  },
  {
    id: 2,
    post: 'some post2',
    title: 'title 2',
    author: 'Ivanov',
    comments: [
      {
        id: 1.1,
        comment: 'some comment1',
        title: 'title 1',
        author: 'Rimus'
      },
      {
```

```

        id: 1.2,
        comment: 'some comment2',
        title: 'title 2',
        author: 'Uncle'
    },
    {
        id: 1.3,
        comment: 'some comment3',
        title: 'title 3',
        author: 'Rimus'
    }
]
},
{
    id: 3,
    post: 'some post3',
    title: 'title 3',
    author: 'Rimus'
},
{
    id: 4,
    post: 'some post4',
    title: 'title 4',
    author: 'Uncle'
}
]

```

```

function getQuntityPostsByAuthor (listOfPosts1, 'Rimus') {
    // your code
}
result -> post -1, comments – 3

```

4.Напишите функцию кеш

```

let complexFunction = function (arg1,arg2) {
    rerurn arg1+arg2; //just for example (any logic can be here)
}

let cachedFunc = cache(complexFunction);

```

```
cachedFunc('foo', 'bar') // complexFunction должна выполниться  
cachedFunc('foo', 'bar') // complexFunction не должна выполняться  
                        // снова, должен вернуться кеш  
cachedFunc('foo', 'baz') // complexFunction должна выполниться  
//потому что метод не вызывался раньше с этими аргументами
```