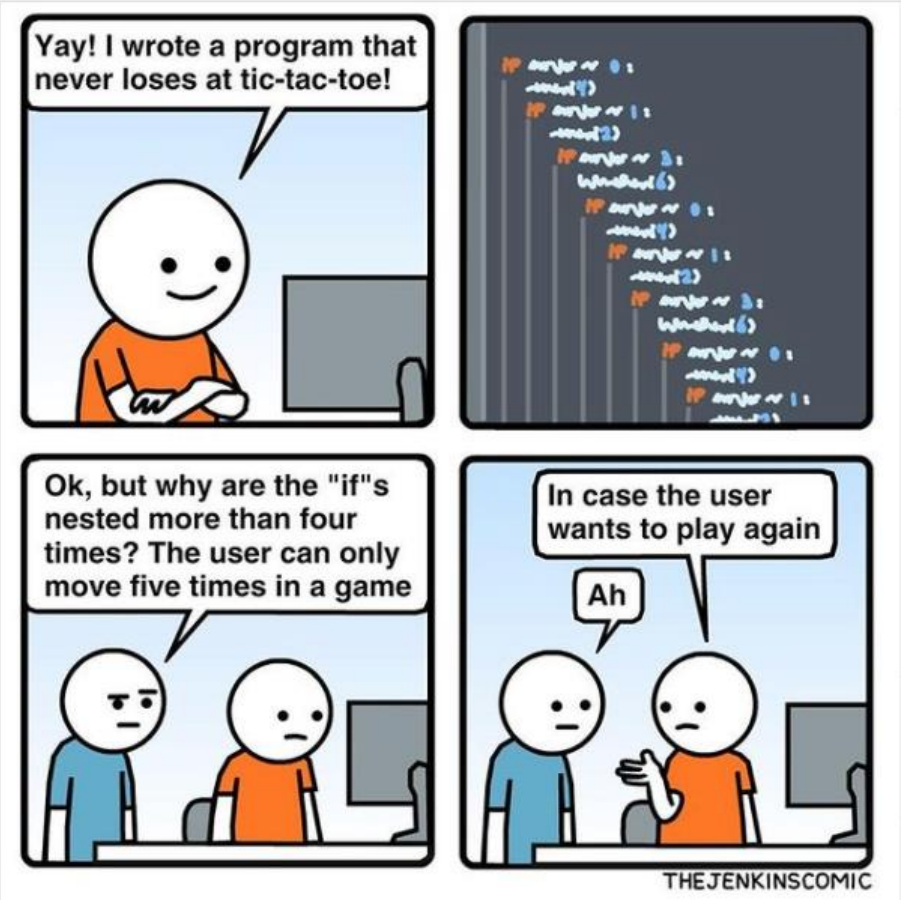


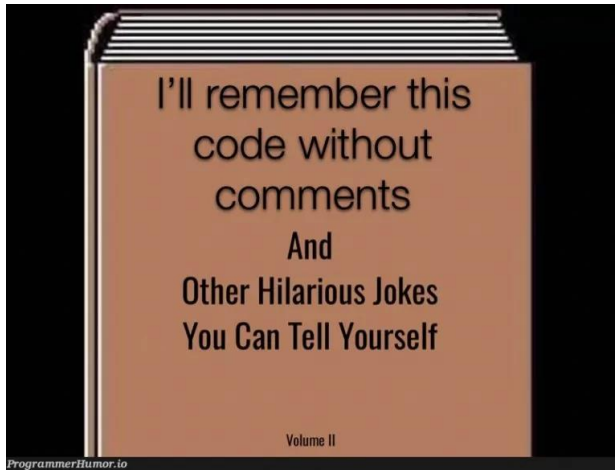

```
public static int najvacsie7() {
    var res = 0;
    for (var c1 = 9; c1 > 0; c1--) {
        for (var c2 = 9; c2 > 0; c2--){
            if(c1 == c2) continue;
            for (var c3 = 9; c3 > 0; c3--){
                if(c1 == c3 || c3 == c2) continue;
                for (var c4 = 9; c4 > 0; c4--){
                    if(c1 == c4 || c4 == c2 || c4 == c3) continue;
                    for (var c5 = 9; c5 > 0; c5--){
                        if(c1 == c5 || c5 == c2 || c5 == c3 || c5 == c4) continue;
                        for (var c6 = 9; c6 > 0; c6--){
                            if(c1 == c6 || c6 == c2 || c6 == c3 || c6 == c4 || c6 == c5) continue;
                            for (var c7 = 9; c7 > 0; c7--){
                                if(c1 == c7 || c7 == c2 || c7 == c3 || c7 == c4 || c7 == c5 || c7 == c6) continue;
                                  var num = String.valueOf(c1) + String.valueOf(c2)+
                                          String.valueOf(c3) + String.valueOf(c4) + String.valueOf(c5)+
                                          String.valueOf(c6) + String.valueOf(c7);
                                  var number = Integer.parseInt(num);
                                  if(number % c1 == 0 && number % c2 == 0 &&
                                          number % c3 == 0 && number % c4 == 0 &&
                                  number % c5 == 0 && number % c6 == 0 && number % c7 == 0){
                                      return number;
                                  }
```

Takto sa to komentuje ☺

```
public class Sedemciferne {

    //Napísal som jednu funkciu, aby som sa nemusel používať celkom 2 cykly, jeden v každej funkcii.
    // Preto pridám komentáre, aby bol kód pochopiteľnejší
    public  static boolean isdivide_by_all_and_all_different(int a){
        int[] numbers = {1,2,3,4,5,6,7,8,9}; // pole na kontrolu unikátnych znakov
        int x = a;
        int j;
        for (int i = 1; i <= 7; i++){
            j = x%10;
            if (j==0) return false;  //aby sa nevyskytla chyba s indexom -1.A tiež podľa podmienky nemáme povolenú číslicu 0

            if (numbers[j-1] == 10) return false; // Ak sme už použili túto hodnotu
            else numbers[j-1] = 10;//Označiť ako použité

            if(a%j!=0) return false;//Ak nie je deliteľná cifrou, z ktorej sa skladá
            x=x/10;
        }
        return true;
    }
```

boolean

Why 10?



If (gonnaOverflow()) don’t;

```
public class Zaujimave {
    public static int pocetZaujimavychInt(int n) {
        if (n == Integer.MAX_VALUE) {
            return 214748364;
        }
        return (n + 1) / 10;
    }
    public static long pocetZaujimavychLong(long n) {
        if (n == Long.MAX_VALUE) {
            return 922337203685477580L;
        }
        return (n + 1) / 10;
    }
    public static long pocetBinarneZaujimavychLong(long n) {
        if (n == Long.MAX_VALUE) {
            return 2305843009213693952L;
        }
```

```
public static long pocetBinarneZaujimavychLong(
    if (n == 9223372036854775807L){
        return 2305843009213693952L;
    }
    if (n%4 != 0){
        return (n + 1) >> 2;
    }
    return (n >> 2);
}
```

```
public class Zaujimave {
    public static int pocetZaujimavychInt(int n) {
        if (n == Integer.MAX_VALUE) {
            return n / 10;
        }
        return (n + 1) / 10;
    }

    public static long pocetZaujimavychLong(long n) {
        if (n == Long.MAX_VALUE) {
            return n / 10;
        }
        return (n + 1) / 10;
    }
```

1 hodina debugovania ušetrí 5minút písania testov.

```
public static void main(String args[]) {
    System.out.println(sucet(1, 1000));
    System.out.println(sucet(10,10));
    System.out.println(sucet(0, Integer.MAX_VALUE));
    System.out.println(sucet(0, Integer.MAX_VALUE, 1));
    System.out.println(sucet(1, 11, 5));
    System.out.println(sucet(1, 10, 5));
    System.out.println(sucet(1, 10, 2));
}
```

Writing unit tests

Pros:	Cons:
-they'll improve the quality of my code -it'll take like 10 mins max -literally everyone says that I should	-i don't wanna

Find 1 diffrence

```
public static long sucet(int a, int b) {
    long n = (long) b - (long) a + 1;
```

```
public static long sucet(int a, int b) {
    long n = (long) b - a + 1;
```

```java
public class Sedemciferne {
  public static int najvacsie7() {
      return 9867312;
  }

  public static void main(String[] args) {
      System.out.println(najvacsie7());
  }
}
```

```java
public class Sedemciferne {
  public static int najvacsie7() {
      // najrpv si skusime urcit, ktore cifry dame do sedemciferneho cisla
      // je zjavne, ze 5ku tam nemozme dat lebo na konci musi byt delitelny 2 a nemoze mat 0
      // potom scitame 7 najvacsich zvysnich cifier dostaneme 39 - tym ze cislo musi byt delitelne 9 tak skusime sucet cifier zmensit na 36
      // takto sme dostali cifri 9,8,7,6,3,2,1
      // posledne trojcislie musi byt delitelne 8 a zaroven 2kou taketo najvacsie cislo je 9 876 321, ale bohuzial toto cislo ne je delitelne 7ckou
      // Skusime znizit cislo (6ku dame do posledneho trojcisla). A zaroven davame pozor aby cisla splnali vlastnosti, ze nech je delitelne 8 a zaroven 2kou
      // Taketo cisla su:
      // 9 873 216
      // 9 872 136
      // 9 871 632
      // Ani jedno z tychto cisel nie je delitelne 7ckou
      // tak ideme dalej a skusime najvacsie cislo kde sedmicka nie je na svojom mieste a splname delitelnost 8mi a 2ma:
      // Take cislo je 9 867 312 -> Toto cislo je delitelne siedmimi Huraaaaa
      return 9_867_312;
  }

  public static void main(String[] args) {
  }
}
```

```java
public class Sedemciferne {
    public static int najvacsie7(){
        for (int i = 9999998; i > 0; i--) {
            int f = i % 10;
            if (f == 0) {
                continue;
            }
            int s = i % 100 / 10;
            if (s == 0) {
                continue;
            }
            int t = i % 1000 / 100;
            if (t == 0) {
                continue;
            }
            int f4 = i % 10000 / 1000;
            if (f4 == 0) {
                continue;
            }
            int f5 = i % 100000 / 10000;
            if (f5 == 0) {
                continue;
            }
            int s6 = i % 1000000 / 100000;
            if (s6 == 0) {
                continue;
            }
            int s7 = i % 10000000 / 1000000;
            if (s7 == 0) {
                continue;
            }
```

To be continued...