

## Задачи на логику, на английском - уно

---

1. When squirrels get together for a party, they like to have cigars. A squirrel party is successful when the number of cigars is between 40 and 60, inclusive. Unless it is the weekend, in which case there is no upper bound on the number of cigars. Return true if the party with the given values is successful, or false otherwise.

```
cigarParty(30, false) → false  
cigarParty(50, false) → true  
cigarParty(70, true) → true
```

2. You are driving a little too fast, and a police officer stops you. Write code to compute the result, encoded as an int value: 0=no ticket, 1=small ticket, 2=big ticket. If speed is 60 or less, the result is 0. If speed is between 61 and 80 inclusive, the result is 1. If speed is 81 or more, the result is 2. Unless it is your birthday -- on that day, your speed can be 5 higher in all cases.

```
caughtSpeeding(60, false) → 0  
caughtSpeeding(65, false) → 1  
caughtSpeeding(65, true) → 0
```

3. The number 6 is a truly great number. Given two int values, a and b, return true if either one is 6. Or if their sum or difference is 6. Note: the function Math.abs(num) computes the absolute value of a number.

```
love6(6, 4) → true  
love6(4, 5) → false  
love6(1, 5) → true
```

4. Return true if the given non-negative number is 1 or 2 more than a multiple of 20. See also: Introduction to Mod

```
more20(20) → false  
more20(21) → true  
more20(22) → true
```

5. Given a non-negative number "num", return true if num is within 2 of a multiple of 10. Note: (a % b) is the remainder of dividing a by b, so (7 % 5) is 2. See also: Introduction to Mod

```
nearTen(12) → true  
nearTen(17) → false  
nearTen(19) → true
```

6. We are having a party with amounts of tea and candy. Return the int outcome of the party encoded as 0=bad, 1=good, or 2=great. A party is good (1) if both tea and candy are at least 5. However, if either tea or candy is at least double the amount of the other one, the party is great (2). However, in all cases, if either tea or candy is less than 5, the party is always bad (0).

```
teaParty(6, 8) → 1  
teaParty(3, 8) → 0  
teaParty(20, 6) → 2
```

---

7. Given three ints, a b c, return true if it is possible to add two of the ints to get the third.

```
twoAsOne(1, 2, 3) → true
twoAsOne(3, 1, 2) → true
twoAsOne(3, 2, 2) → false
```

8. Given three ints, a b c, return true if two or more of them have the same rightmost digit. The ints are non-negative. Note: the % "mod" operator computes the remainder, e.g. 17 % 10 is 7.

```
lastDigit(23, 19, 13) → true
lastDigit(23, 19, 12) → false
lastDigit(23, 19, 3) → true
```

9. Given two int values, return whichever value is larger. However if the two values have the same remainder when divided by 5, then the return the smaller value. However, in all cases, if the two values are the same, return 0. Note: the % "mod" operator computes the remainder, e.g. 7 % 5 is 2.

```
maxMod5(2, 3) → 3
maxMod5(6, 2) → 6
maxMod5(3, 2) → 3
```

10. You have a blue lottery ticket, with ints a, b, and c on it. This makes three pairs, which we'll call ab, bc, and ac. Consider the sum of the numbers in each pair. If any pair sums to exactly 10, the result is 10. Otherwise if the ab sum is exactly 10 more than either bc or ac sums, the result is 5. Otherwise the result is 0.

```
blueTicket(9, 1, 0) → 10
blueTicket(9, 2, 0) → 0
blueTicket(6, 1, 4) → 10
```

11. You and your date are trying to get a table at a restaurant. The parameter "you" is the stylishness of your clothes, in the range 0..10, and "date" is the stylishness of your date's clothes. The result getting the table is encoded as an int value with 0=no, 1=maybe, 2=yes. If either of you is very stylish, 8 or more, then the result is 2 (yes). With the exception that if either of you has style of 2 or less, then the result is 0 (no). Otherwise the result is 1 (maybe).

```
dateFashion(5, 10) → 2
dateFashion(5, 2) → 0
dateFashion(5, 5) → 1
```

12. Given 2 ints, a and b, return their sum. However, sums in the range 10..19 inclusive, are forbidden, so in that case just return 20.

```
sortaSum(3, 4) → 7
sortaSum(9, 4) → 20
sortaSum(10, 11) → 21
```

13. Given a number n, return true if n is in the range 1..10, inclusive. Unless outsideMode is true, in which case return true if the number is less or equal to 1, or greater or equal to 10.

---

```
in1To10(5, false) → true
in1To10(11, false) → false
in1To10(11, true) → true
```

14. Return true if the given non-negative number is a multiple of 3 or 5, but not both. Use the % "mod" operator -- see Introduction to Mod

```
old35(3) → true
old35(10) → true
old35(15) → false
```

15. Given 2 ints, a and b, return their sum. However, "teen" values in the range 13..19 inclusive, are extra lucky. So if either value is a teen, just return 19.

```
teenSum(3, 4) → 7
teenSum(10, 13) → 19
teenSum(13, 2) → 19
```

16. Given a string str, if the string starts with "f" return "Fizz". If the string ends with "b" return "Buzz". If both the "f" and "b" conditions are true, return "FizzBuzz". In all other cases, return the string unchanged. (See also: FizzBuzz Code)

```
fizzString("fig") → "Fizz"
fizzString("dib") → "Buzz"
fizzString("fib") → "FizzBuzz"
```

17. Given three ints, a b c, return true if b is greater than a, and c is greater than b. However, with the exception that if "bOk" is true, b does not need to be greater than a.

```
inOrder(1, 2, 4, false) → true
inOrder(1, 2, 1, false) → false
inOrder(1, 1, 2, true) → true
```

18. Given three ints, a b c, return true if one of them is 10 or more less than one of the others.

```
lessBy10(1, 7, 11) → true
lessBy10(1, 7, 10) → false
lessBy10(11, 1, 7) → true
```

19. You have a red lottery ticket showing ints a, b, and c, each of which is 0, 1, or 2. If they are all the value 2, the result is 10. Otherwise if they are all the same, the result is 5. Otherwise so long as both b and c are different from a, the result is 1. Otherwise the result is 0.

```
redTicket(2, 2, 2) → 10
redTicket(2, 2, 1) → 0
redTicket(0, 0, 0) → 5
```

20. Given two ints, each in the range 10..99, return true if there is a digit that appears in both numbers, such as the 2 in 12 and 23. (Note: division, e.g. `n/10`, gives the left digit while the `% "mod"` `n%10` gives the right digit.)

```
shareDigit(12, 23) → true
shareDigit(12, 43) → false
shareDigit(12, 44) → false
```

21. The squirrels in Palo Alto spend most of the day playing. In particular, they play if the temperature is between 60 and 90 (inclusive). Unless it is summer, then the upper limit is 100 instead of 90. Given an int temperature and a boolean `isSummer`, return true if the squirrels play and false otherwise.

```
squirrelPlay(70, false) → true
squirrelPlay(95, false) → false
squirrelPlay(95, true) → true
```

22. Given a day of the week encoded as 0=Sun, 1=Mon, 2=Tue, ...6=Sat, and a boolean indicating if we are on vacation, return a string of the form "7:00" indicating when the alarm clock should ring. Weekdays, the alarm should be "7:00" and on the weekend it should be "10:00". Unless we are on vacation -- then on weekdays it should be "10:00" and weekends it should be "off".

```
alarmClock(1, false) → "7:00"
alarmClock(5, false) → "7:00"
alarmClock(0, false) → "10:00"
```

23. We'll say a number is special if it is a multiple of 11 or if it is one more than a multiple of 11. Return true if the given non-negative number is special. Use the `% "mod"` operator -- see Introduction to Mod

```
specialEleven(22) → true
specialEleven(23) → true
specialEleven(24) → false
```

24. Return true if the given non-negative number is 1 or 2 less than a multiple of 20. So for example 38 and 39 return true, but 40 returns false. See also: Introduction to Mod

```
less20(18) → true
less20(19) → true
less20(20) → false
```

25. Your cell phone rings. Return true if you should answer it. Normally you answer, except in the morning you only answer if it is your mom calling. In all cases, if you are asleep, you do not answer.

```
answerCell(false, false, false) → true
answerCell(false, false, true) → false
answerCell(true, false, false) → false
```

26. Given an int `n`, return the string form of the number followed by "!". So the int 6 yields "6!". Except if the number is divisible by 3 use "Fizz" instead of the number, and if the number is divisible by 5 use "Buzz", and if divisible by both 3 and 5, use

"FizzBuzz". Note: the % "mod" operator computes the remainder after division, so 23 % 10 yields 3. What will the remainder be when one number divides evenly into another? (See also: FizzBuzz Code and Introduction to Mod)

```
fizzString2(1) → "1!"  
fizzString2(2) → "2!"  
fizzString2(3) → "Fizz!"
```

27. Given three ints, a b c, return true if they are in strict increasing order, such as 2 5 11, or 5 6 7, but not 6 5 7 or 5 5 7. However, with the exception that if "equalOk" is true, equality is allowed, such as 5 5 7 or 5 5 5.

```
inOrderEqual(2, 5, 11, false) → true  
inOrderEqual(5, 7, 6, false) → false  
inOrderEqual(5, 5, 7, true) → true
```

28. Return the sum of two 6-sided dice rolls, each in the range 1..6. However, if noDoubles is true, if the two dice show the same value, increment one die to the next value, wrapping around to 1 if its value was 6.

```
withoutDoubles(2, 3, true) → 5  
withoutDoubles(3, 3, true) → 7  
withoutDoubles(3, 3, false) → 6
```

29. You have a green lottery ticket, with ints a, b, and c on it. If the numbers are all different from each other, the result is 0. If all of the numbers are the same, the result is 20. If two of the numbers are the same, the result is 10.

```
greenTicket(1, 2, 3) → 0  
greenTicket(2, 2, 2) → 20  
greenTicket(1, 1, 2) → 10
```

30. Given 2 non-negative ints, a and b, return their sum, so long as the sum has the same number of digits as a. If the sum has more digits than a, just return a without b. (Note: one way to compute the number of digits of a non-negative int n is to convert it to a string with String.valueOf(n) and then check the length of the string.)

```
sumLimit(2, 3) → 5  
sumLimit(8, 3) → 8  
sumLimit(8, 1) → 9
```