

# ТЕСТ ПО ИНФОРМАТИКЕ

1. Существует компания, активно следящая за социальной адаптацией своих сотрудников. Человека называют *нелюдимым*, если у него имеется не более трёх знакомых внутри компании. Известно, что у каждого сотрудника организации не менее трёх нелюдимых знакомых. Всего в компании работают  $n$  человек. Требуется посчитать количество нелюдимых сотрудников.

There is a company that actively monitors the social adaptation of its employees. A person is called *unsociable* if they have no more than three acquaintances inside the company. It is known that every employee of the organization has at least three unsociable acquaintances. In total, the company employs  $n$  people. It is required to count the number of unsociable employees.

- (a)  $n$
- (b)  $n/3$
- (c)  $n - 3$
- (d)  $n - 4$
- (e)  $n/3 + 1$

2. Даны рекурсивная функция  $f$ , принимающая на вход натуральное число  $n$ , и функция  $g$ , вызывающая эту функцию  $f$ . Реализации обеих функций приведены ниже. Требуется определить, что выведут на экран последовательные вызовы  $g(33)$  и  $g(34)$ .

The recursive function  $f$  is given, it takes the natural number  $n$  as input; also there is the  $g$  function, which calls  $f$ . Implementations of both functions are given below. It is required to determine what consecutive calls of  $g(33)$  and  $g(34)$  will display on the screen.

```
define function "f" that accepts (integer n) and returns integer:
  if n equals 1 or n equals 2:
    return 1
  else:
    return f(n - f(n - 1)) + f(n - f(n - 2))

define function "g" that accepts (integer n) and returns integer:
  return f(n) mod 4
```

- (a) 0 1
- (b) 1 0
- (c) 3 3

(d) 1 1

(e) 2 1

3. Ниже приведён фрагмент кода. Что выведется на экран по окончании его работы?

Below some code snippet is given. What will be displayed on the screen at the end of its operation?

```
unsigned integer i = 100;
while (i >= 0) {
    print(i);
    print(' ');
    i = i - 1;
}
```

- (a) все целые числа от 100 до 0, разделённые символом пробела / all integers from 100 to 0, separated by a space character
- (b) все целые числа от 100 до 1, разделённые символом пробела / all integers from 100 to 1, separated by a space character
- (c) ничего не выведется / nothing will be displayed
- (d) поведение непредсказуемо / unpredictable behavior
- (e) цикл не закончит свою работу / the loop will not finish its work

4. Пусть мы некоторым образом храним в памяти информацию о броске двух игральных кубиков, то есть, иначе говоря, храним вместе два натуральных числа –  $dice_1$  и  $dice_2$ , диапазон которых – 1-6. Пусть теперь мы совершили несколько бросков подряд, и все они получились различными (различными будем считать любые два броска, кроме бросков вида  $[n, m]$ ,  $[n, m]$ , то есть  $[2, 1]$  и  $[1, 2]$  – разные броски, а вот  $[1, 2]$  и  $[1, 2]$  – одинаковые). Мы хотим разместить результаты нашего эксперимента (пары чисел) в массиве, нумерующемся с нуля, так, чтобы различные броски не попадали в одну и ту же ячейку массива. Как в этом случае должна выглядеть функция, определяющая номер ячейки, в которую отправится очередной бросок?

Let's suppose that we store information about the roll of two dice in memory in some way, that is, in other words, we store together two natural numbers –  $dice_1$  and  $dice_2$ , whose range is 1-6. Suppose now we have made several throws in a row, and they all turned out to be different (we will consider any two throws different, except for throws of the form  $[n, m]$ ,  $[n, m]$ , that is,  $[2, 1]$  and  $[1, 2]$  are different throws, but  $[1, 2]$  and  $[1, 2]$  are the same). We want to add the results of our experiment (pairs of numbers) into an array numbered from zero, in such a way that different throws do not fall into the same cell of the array. In this case, what should the function that determines the number of the cell to which the next roll will go look like?

(a)  $dice_1 + dice_2$

(b)  $dice_1 * dice_2$

- (c)  $dice_1 * 3 + dice_2$
- (d)  $dice_1 * 13 + dice_2$
- (e)  $dice_1 * 5 + dice_2$

5. Логическая (булева) функция называется *самодвойственной*, если выполнено равенство:  $f(x_1, \dots, x_n) = \bar{f}(\bar{x}_1, \dots, \bar{x}_n)$ . Пусть даны две функции  $f = x_1x_2 \vee x_2x_3 \vee x_3x_1$  и  $g = x_1 \oplus x_2 \oplus x_3 \oplus 1$ , где  $\oplus$  – исключающее «ИЛИ». Необходимо определить, являются ли они самодвойственными.

A Boolean function is called *self-dual* if the following equality holds:  $f(x_1, \dots, x_n) = \bar{f}(\bar{x}_1, \dots, \bar{x}_n)$ . Now two functions are given:  $f = x_1x_2 \vee x_2x_3 \vee x_3x_1$  and  $g = x_1 \oplus x_2 \oplus x_3 \oplus 1$ , where  $\oplus$  is an exclusive «OR». You are asked to determine whether they are self-dual or not.

- (a)  $f$  – является,  $g$  – не является /  $f$  – self-dual,  $g$  – not
- (b)  $f$  – не является,  $g$  – является /  $f$  – not,  $g$  – self-dual
- (c) обе являются / both are self-dual
- (d) обе не являются / both are not

6. Рассмотрим абстрактную вычислительную систему под названием *машина с неограниченными регистрами*. Она состоит из бесконечно длинной ленты, разделённой на ячейки – регистры, в которые можно записать любое неотрицательное целое число. Каждый регистр имеет свой номер, нумерация осуществляется с единицы. Для функции  $f(x_1, x_2, \dots, x_n)$  значения аргументов  $x_1, x_2, \dots, x_n$  записываются подряд в регистры с номерами  $1, 2, \dots, n$ . Вот некоторые из операций, которые умеет делать такая машина:

- 1)  $Z(n)$  – обнуление регистра с номером  $n$ ;
- 2)  $S(n)$  – увеличение значения регистра с номером  $n$  на единицу;
- 3)  $I(m, n, q)$  – условный переход: происходит сравнение значений, записанных в регистрах с номерами  $m$  и  $n$ ; если эти значения равны, то машина переходит к строке программы с номером  $q$ , иначе машина просто переходит к следующей строке программы.

Считается, что машина завершила свою работу, если она достигла последней строки программы, и эта строка не является условным переходом, либо если встретилась команда условного перехода со значением  $q$ , превышающим количество строк программы. Результатом работы считается содержимое первого регистра.

Вам дана некоторая программа машины с неограниченными регистрами, которая вычисляет функцию от двух аргументов  $f(x, y)$  (непроинициализированные регистры положим равными 0). Что это за функция?

Let's consider an abstract computing system called *machine with unlimited registers*. It consists of an infinitely long tape divided into cells – registers into which any non-negative integer can be written. Each register has its own number, numbering is carried out from one. For the function  $f(x_1, x_2, \dots, x_n)$  the values of the arguments  $x_1, x_2, \dots, x_n$  are written in a row to registers with numbers  $1, 2, \dots, n$ . Here are some operations that such a machine can do:

- 1)  $Z(n)$  – zeroing the register with the number  $n$ ;

- 2)  $S(n)$  – increasing the value of the register with the number  $n$  by one;
- 3)  $I(m, n, q)$  – conditional transition: the values recorded in the registers with the numbers  $m$  and  $n$  are compared; if these values are equal, then the machine goes to the program line with the number  $q$ , otherwise the machine simply goes to the next line of the program.

It is considered that the machine has completed its work if it has reached the last line of the program, and this line is not a conditional transition, or if a conditional transition command has been encountered with a value of  $q$  exceeding the number of lines of the program. The result of the work is the content of the first register.

You are given some machine with unlimited registers program that computes a function from two arguments  $f(x, y)$  (let's put the uninitialized registers equal to 0). What is this function?

1.  $I(2, 3, 5)$
2.  $S(1)$
3.  $S(3)$
4.  $I(1, 1, 1)$

(a)  $f(x, y) = x + y + 1$

(b)  $f(x, y) = x + y - 1$

(c)  $f(x, y) = x * y$

(d)  $f(x, y) = x$

(e)  $f(x, y) = x + y$

(f)  $f(x, y) = y$