

**Олимпиада школьников по информатике и компьютерной безопасности
(2008 год)**

9 класс

Решение задачи № 1

Решение. Всего в записи числа используется 30 бит. Известно, что это запись мобильного номера телефона. Любой такой номер содержит 10 цифр не считая префикса +7 или 8. Число 0 и 1 в записи равно 31 и нацело не делится на 10. Следовательно, для записи различных цифр использовались последовательности различной длины. Однако друг оставил подсказку в виде скобок и чёрточек в привычной записи мобильного номера. Используя представление цифр в двоичной системе попытаемся восстановить номер. Начнём с последовательности, стоящей в скобках. Здесь записаны 3 цифры. Предположим, что это двоичная запись числа. Для того чтобы перевести число, следует воспользоваться многочленом разложения в следующем виде:

$$A = \sum_{i=0}^{n-1} k_i * 2^i$$

где A – число в десятичной системе счисления, n – количество цифр в числе, которое нужно перевести, i – позиция цифры в числе, которое нужно перевести, k_i – цифра, которая находится на i -ой позиции в числе, которое нужно перевести. В результате получаем 922. Следующие три цифры образуют число 145. Оставшиеся две последовательности нулей и единиц соответствуют числам – 56 и 98. В итоге номер друга – (914)-145-56-98.

Ответ: Номер друга – (922)-145-56-98.

Решение задачи № 2

Решение задачи связано с необходимостью проверки входных данных на их соответствие имеющимся условиям. В данном случае в качестве входных данных выступает количество гостей, которое придет на праздник. Поэтому после пункта «(с) – встретить гостей» должна следовать проверка на то, сколько гостей пришло, и если пришло больше, чем 12, нужно предусмотреть какие-то действия.

Решение задачи № 3

Решение задачи связано с подсчетом возможного количества вариантов пароля. Затем это количество умножается на время одной попытки, что дает максимальное время, которое потребуется на подбор пароля.

Для того чтобы подсчитать количество вариантов, воспользуемся следующей формулой:

$$N = 6 * C_7^2 * 5! = 6! * C_7^2 = 6! * \frac{7!}{5! * 2!} = 6 * 3 * 4 * 5 * 6 * 7 = 2520$$

Теперь, чтобы подсчитать время:

$$T = 2520 * 5 = 12600 \text{ сек.} = 210 \text{ мин.} = 2 \text{ часа } 30 \text{ минут.}$$

Решение задачи № 4

В данной задаче необходимо найти некую закономерность, которая бы позволила определить, каким образом кодируется ноль и единица. Таких закономерностей может быть несколько. Из полученных вариантов нужно отобрать тот, в котором коды получаемых ASCII – символов будут находиться в интервале от 32 до 127.

В этом случае останется следующая закономерность. В данном html-файле используется таблица. В этой таблице 42 строки. Каждая из этих строк либо заканчивается служебными тегами таблицы (<table>, <tr>, <td>, </td>, </tr>, </table>), что означает ноль, либо нет – это означает 1. Коды всех получаемых символов лежат в интервале от 32 до 127. Таким образом, мы установили искомую закономерность. Значит, в нашем пароле будет использоваться 6 символов.

Решение задачи № 5

В данной задаче используется операция XOR. Данная операция является побитовой и реализуется следующим образом:

$$0 \text{ XOR } 0 = 0$$

$$0 \text{ XOR } 1 = 1$$

$$1 \text{ XOR } 0 = 1$$

$$1 \text{ XOR } 1 = 0$$

Для того чтобы удостовериться в том, что пакет был передан правильно, необходимо рассчитать контрольную сумму для переданного пакета и сравнить ее с той контрольной суммой, которая была в переданном пакете. Если контрольные суммы совпадут, значит, пакет был передан правильно. В противном случае пакет передан не верно.

$$01110111 \text{ XOR } 11100011 \text{ XOR } 10001111 \text{ XOR } 10101010 \text{ XOR } 00100010 \text{ XOR } 01011100 \text{ XOR } 11000110 \text{ XOR } 11100011 = 111101010$$

В пакете была передана следующая контрольная сумма – 11000100. Она, как можно видеть, не совпадает с той, которая была рассчитана. Значит, пакет передан неверно.

Решение задачи № 6

Ключевым моментом решения данной задачи является тот факт, что программисты использовали одну и ту же случайную последовательность для применения операции XOR. Это означает, что если данную операцию проводить четное число раз, то последовательность, полученная в итоге, не будет отличаться от исходной последовательности.

Например, пусть исходная последовательность выглядит так – 01010001 11101011. Пусть случайная последовательность выглядит так – 11100010 10111100. Тогда после применения операции XOR один раз мы получим -
01010001 11101011 XOR 11100010 10111100 = 10110011 01010111.

После проведения операции XOR второй раз получим

$$10110011 01010111 \text{ XOR } 11100010 10111100 = 01010001 11101011.$$

Видно, что результирующая последовательность после проведения второй операции не отличается от исходной последовательности. Поэтому ошибка программистов заключается в том, что N не может быть четным.