

Условия задач заключительного этапа

Задача 1. Пин-код

У Альберта в качестве пин-кода от банковской карты используется такое простое четырехзначное число, для которого сумма первых трёх цифр равна последней цифре. После 10 неправильных попыток ввода пин-кода карта блокируется. Можно ли гарантированно ввести правильный пароль и не заблокировать карту, если Альберт его забыл? Ответ обоснуйте.

Решение

Простое четырехзначное число лежит в диапазоне от 1000 до 9999. Именно в этом диапазоне необходимо найти простые числа, удовлетворяющие условию "сумма первых трёх цифр равна последней цифре". Если представить простое число в виде $abcd$, то условие будет выглядеть следующим образом:

$$a+b+c=d$$

Для ускорения процесса поиска таких чисел можно написать программу.

Пример функции проверки простого числа на языке C.

```
// ФУНКЦИЯ проверки, является ли число простым
// INPUT - NUM - целое число
// RETURN - TRUE - число простое
//          FALSE - число непростое
bool is_prost(int num)
{
    int d = 2;                // делитель числа
    // цикл по делителю от 2-х до num/2
    while (d < num / 2)
    {
        // если остаток от деления num на d
        // нулевой, число - непростое, return false
        if (num % d == 0)
            return false;
        d++;                // увеличиваем делитель на 1
    }
    return true;
}
```

Пример функции проверки условия $a+b+c=d$ на языке C.

```
// ФУНКЦИЯ проверки условия a+b+c=d
// INPUT - NUM - целое число
// RETURN - TRUE - число удовлетворяет условию
//          FALSE - число НЕ удовлетворяет условию
bool check(int num)
{
    int a = num / 1000;        // 1-й разряд числа NUM
    int b = (num / 100) % 10; // 2-й разряд числа
                                NUM
    int c = (num / 10) % 10; // 3-й разряд числа NUM
    int d = num % 10;         // 4-й разряд числа NUM
```

```

    if (a + b + c == d)
        return true;
    return false;
}

```

Пример функции `main()`, выводящей на экран простые числа, удовлетворяющие условию.

```

int main()
{
    // Перебор всех 4-значных чисел от 1000 до 9999
    // и вывод на экран только простых и
    // удовлетворяющих условию check()
    for (int n = 1000; n < 10000; n++)
        if (is_prost(n) && check(n))
            printf("%d\n", n);
    return 0;
}

```

Результат выполнения программы выводит следующие числа:

```

1427
1607
2237
2417
3137
3407
4127
4217

```

Всего простых чисел, подходящих под условие – 8. Значит, если Альберт забыл пин-код, он сможет перебрать все возможные варианты без блокирования карты.

Ответ: да, подходящих под условия пин-кодов всего 8.

Задача 2. Тайное послание

В феврале 1890 г. начальник сыскной полиции г. Санкт-Петербурга Иван Дмитриевич Путилин проводил расследование очередного преступления. Имелся подозреваемый, но не хватало доказательств. При обыске в номере гостиницы, где проживал предполагаемый преступник, Иван Дмитриевич обнаружил на столе три предмета: листок бумаги с нарисованной от руки таблицей, книгу и перевёрнутую вверх дном кружку с надписью «13» на доньшке.

После недолгих размышлений Ивану Дмитриевичу удалось получить необходимую информацию для раскрытия преступления. Удастся ли это Вам? Ниже представлена таблица, нарисованная на найденном листе бумаги. Книга расположена в отдельном файле.

10, 4, 2	12, 2, 96	7, 14, 95	9, 2, 65						
		6, 8, 1	42, 11, 86						
				28, 11, FF					
					20, 5, 4				
				12, 7, 14	8, 4, 3				

Электронные материалы: электронная книга Стивенсон Роберт Луис - *Остров сокровищ* (96 стр.)
(http://v-olymp.ru/olmp_it/docs/2018/tasks/9-10/2/Стивенсон_Роберт_Луис_-_Остров_сокровищ.pdf).

Решение

Таблица состоит из 98 ячеек, столько же страниц в книге. Нетрудно догадаться, что каждая ячейка соответствует странице в книге.

Можно предположить, что первое число указывает на номер строки сверху, а второе - на номер слова в строке слева.

Третье число в ячейках не повторяется и лежит в диапазоне от 1 до 98, кроме значения FF. Можно предположить, что третье число указывает на следующую ячейку, а ячейка FF – это последняя ячейка.

Остается вопрос: с какой ячейки/страницы начинать?

В задании указано число «13», значит первое слово на 13-й странице, информация об этом слове содержится в ячейке 13: 6-я строка, 8-е слово – «ЗОЛОТО».

Следующее слово на странице 1: 10-я строка, 4-е слово – «ЛЕЖИТ».

Следующая страница 2 и т.д. (см. таблицу).

<i>Страница/ ячейка</i>	<i>№ строки</i>	<i>№ слова</i>	<i>Слово</i>
13	6	8	ЗОЛОТО
1	10	4	ЛЕЖИТ
2	12	2	В
96	8	4	СУНДУКЕ
3	7	14	В
65	28	11	КАЮТЕ
14	39	11	ТРУП
86	20	5	СПРЯТАН
4	9	2	В
95	12	7	ПЕЩЕРЕ

Ответ: Золото лежит в сундуке в каюте. Труп спрятан в пещере.

Задача 3. Скрытое сообщение

Взаимодействие между агентами осуществляется по каналу связи, позволяющему последовательно передавать несколько текстовых сообщений произвольного ненулевого размера, кратного 1 Кб. В ходе осуществления очередного сеанса связи было отправлено 3 сообщения общим размером 16 Кб. Для однозначного определения размеров и места каждого сообщения в сеансе связи агенту на приемной стороне потребовалось дополнительно сообщить значение, равное произведению их размеров, и тот факт, что сообщение наибольшего размера было отправлено первым. Помогите агенту определить размеры полученных сообщений с учетом порядка их отправки.

Решение

Рассмотрев все возможные комбинации длин сообщений (в сумме которые дают 16), можно заметить, что в большинстве из них нельзя однозначно определить последовательность сообщений, даже обладая информацией о том, что максимальной длины сообщение идет первым. Полученной информации будет достаточно только в том случае, когда два других сообщения будут обладать одинаковой длиной. Под такие критерии подпадают следующие наборы:

$$150=6*5*5$$

$$128=8*4*4$$

$$58=12*2*2$$

$$14=14*1*1$$

$$90=10*3*3$$

Ответ: 6*5*5 или 8*4*4 или 12*2*2 или 14*1*1 или 10, 3, 3.

Задача 4. ZIP

В ходе исследования компьютера одного из сотрудников организации были обнаружены подозрительные файлы. Предварительный анализ показал, что среди них есть файлы, в которых сотрудник хранил пароли к данным в определенном формате:

{XXXX-XXXX-XXXX-XXXX}

Непроанализированным остался единственный файл HowTo.docx. Помогите найти в этом файле возможный пароль.

Электронные материалы: файл HowTo.docx

(http://v-olymp.ru/olmp_it/docs/2018/tasks/9-10/4/HowTo.docx),

файл Proc.zip (http://v-olymp.ru/olmp_it/docs/2018/tasks/9-10/4/Proc.zip).

Решение

Можно обратить внимание, что первые байты файла HowTo.docx совпадают с первыми байтами файла Proc.zip. Это означает, что файл формата .DOCX представляет собой zip-архив.

```

00000000: 50 4B 03 04 14 00 06 00|08 00 00 00 21 00 E7 21 |PK.....!.z!
00000010: 07 5D 70 01 00 00 07 05|00 00 13 00 08 02 5B 43 |.]p...4.....[C
00000020: 6F 6E 74 65 6E 74 5F 54|79 70 65 73 5D 2E 78 6D |ontent_Types].xm
00000030: 6C 20 A2 04 02 28 A0 00|02 00 00 00 00 00 00 00 |l ŷ. ( . ....
00000040: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000050: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000060: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000070: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....
00000080: 00 00 00 00 00 00 00 00|00 00 00 00 00 00 00 00 |.....

```

Рисунок 1 – Содержимое файла HowTo.docx

```

00000000: 50 4B 03 04 14 00 02 00|08 00 07 70 52 4D 83 DF |PK.... .rRMгЯ
00000010: 2A D4 8B 00 00 00 00 00|00 00 08 00 00 50 72 |*Ф<...#.....Pr
00000020: 6F 63 2E 74 78 74 3D 8E|41 0E 82 30 10 45 AF F2 |oc.txt=ќА.,0.EЇт
00000030: C3 1A 3D 86 6B 17 9E DA|9D 37 18 4B 47 4A A1 53 |Г.=†k.ћЪк7.KGJŸS
00000040: 66 1A 43 08 48 EC 46 97|2F 79 3F EF DF A5 F3 04 |f.C.HMF-/y?няГу.
00000050: B7 41 2C 8E C7 42 6C AC|7B 0B 13 E8 24 AF 4A E8 |-А,ќ3B1-ζ...и$ЇЖи
00000060: 29 CB 94 82 ED 30 8E 65|65 86 0A FB 90 38 05 AE |)л",н0ћее†.ыћ8.0
00000070: 22 45 2C D5 96 24 78 16|CF 06 2F 6A 25 57 74 F0 |"E,x-Ѕx.П./j%Wtp
00000080: 81 46 E9 18 CD E3 BF 80|E2 72 BB 9E 94 67 C7 B3 |ѓFй.HгїЪвр>>ћ"q3i

```

Рисунок 2 – Содержимое файла Proc.zip

Переименовав файл HowTo.docx в HowTo.zip можно его открыть любой программой-архиватором (WinZip или WinRar). В результате отобразится содержимое архива в виде некоторого количества файлов и папок.

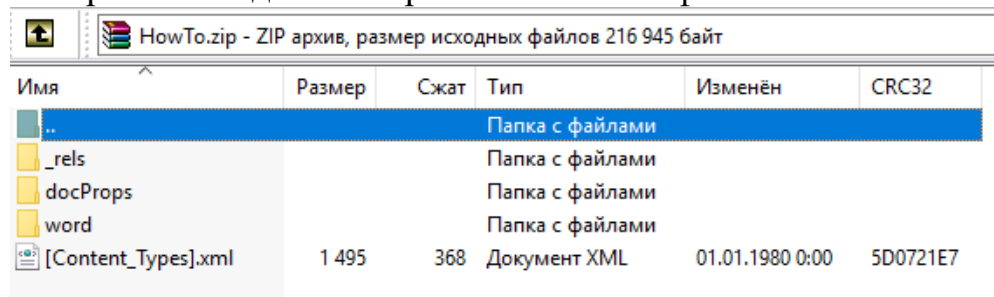
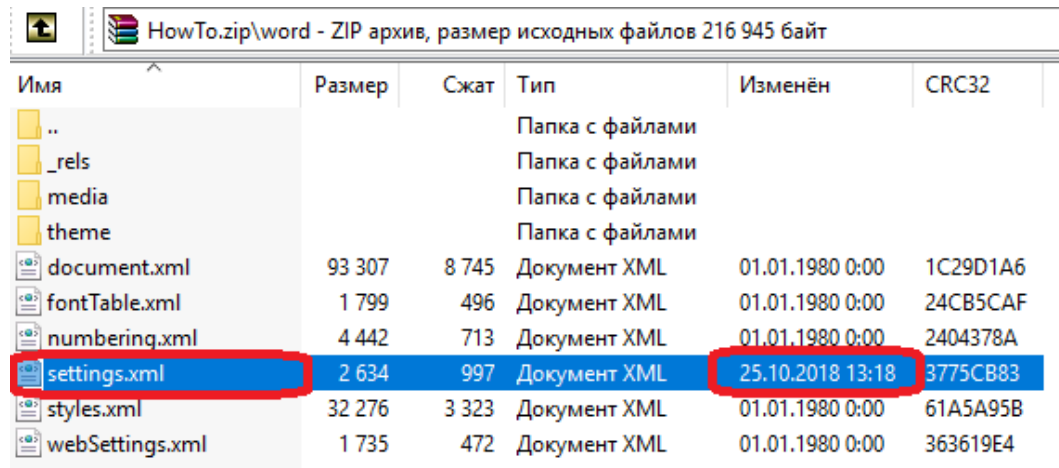


Рисунок 3 – Содержимое файла HowTo.zip

Выполнив поиск данных, подходящих под формат пароля, по всем файлам, содержащимся в архиве, можно определить, что пароль спрятан в файле *settings.xml*.

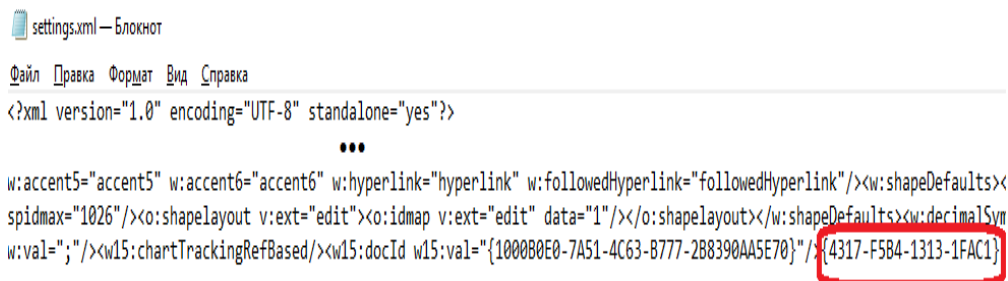
Это можно увидеть одним из способов:

- поиск в содержимом файлов текста формата {XXXX-XXXX-XXXX-XXXXXX},
- анализируя дату изменения файла.



Имя	Размер	Сжат	Тип	Изменён	CRC32
..			Папка с файлами		
_rels			Папка с файлами		
media			Папка с файлами		
theme			Папка с файлами		
document.xml	93 307	8 745	Документ XML	01.01.1980 0:00	1C29D1A6
fontTable.xml	1 799	496	Документ XML	01.01.1980 0:00	24CB5CAF
numbering.xml	4 442	713	Документ XML	01.01.1980 0:00	2404378A
settings.xml	2 634	997	Документ XML	25.10.2018 13:18	3775CB83
styles.xml	32 276	3 323	Документ XML	01.01.1980 0:00	61A5A95B
webSettings.xml	1 735	472	Документ XML	01.01.1980 0:00	363619E4

Рисунок 4 – Измененный файл



```

settings.xml – Блокнот
Файл  Правка  Формат  Вид  Справка
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
...
w:accent5="accent5" w:accent6="accent6" w:hyperlink="hyperlink" w:followedHyperlink="followedHyperlink"/><w:shapeDefaults>
spidmax="1026"/><o:shapelayout v:ext="edit"><o:idmap v:ext="edit" data="1"/></o:shapelayout></w:shapeDefaults><w:decimalSym
w:val=""/><w15:chartTrackingRefBased/><w15:docId w15:val="{1000B0E0-7A51-4C63-B777-2B8390AA5E70}"/>{4317-F5B4-1313-1FAC1}

```

Рисунок 5 – Содержимое файла settings.xml

Ответ: {4317-F5B4-1313-1FAC1}

Задача 5. Подбор пароля

В системе безопасности используется n -разрядный ключ ($0 < n \leq 64$). Для проверки вводимого пользователем ключа вызывается функция *Check()*, которой он передается в качестве параметра. Результатом проверки является число совпадающих разрядов между введенным пользователем и эталонным ключами. Введенный ключ считается правильным, если функция *Check()* вернула значение, равное n . Предложите алгоритм, который гарантированно угадывает значение ключа не более чем за n вызовов функции проверки.

Листинг функции Check()

```

const __int64 KEY;           // эталонный ключ

/* ВХОДНЫЕ ПАРАМЕТРЫ:
* key      - значение проверяемого ключа

```

```

*           (последний разряд - самый младший)
* n         - количество разрядов ключа
* СТОРОННИЕ ПЕРЕМЕННЫЕ:
* KEY       - эталонное значение ключа (объявлено вне функции)
* ВОЗВРАЩАЕМОЕ ЗНАЧЕНИЕ:
* int       - количество верных разрядов ключа key
*/
int Check(__int64 key, int n)
{
    int res = 0;
    for (int i = 0; i < n; i++)
    {
        if ( (key & (0x00000000000000001 << i)) ==
            (KEY & (0x00000000000000001 << i)) )
            res++;
    }
    return res;
}

```

Решение

Как работает функция Check():

- 1) В цикле от 0 до N (длина ключа в битах) происходит сравнение i-го бита проверяемого ключа с эталонным, начиная с крайнего правого.
- 2) Если i-й бит проверяемого ключа совпадает с i-м битом эталонного ключа, счетчик res увеличивается на 1.
- 3) По окончании цикла счетчик содержит число совпавших бит у проверяемого и эталонного ключа. Если счетчик равен N – ключи равны (все биты одинаковые).

По условиям задачи необходимо разработать алгоритм определения эталонного ключа не более, чем за N запусков функции Check().

Предлагается следующее решение:

- 1) Изначально принять ключ, равный 00000...0. Проверяемый бит $k = 0$ (первый).
- 2) Подать его на вход функции Check(). Запомнить результат (res1).
- 3) Инвертировать значение k-го бита (поменять с «0» на «1» или наоборот).
- 4) Вызвать функцию Check() с измененным ключом, поданным на вход. Сравнить результат выполнения функции (res2) с результатом, полученным в п.2 (res1). Если $res2 > res1$, оставить ключ без изменения. Если $res2 < res1$ инвертировать обратно измененный k-й бит.
- 5) Увеличить значение k на 1 ($k = k + 1$).
- 6) Если $k = N - 1$ (последний разряд), то проанализировать результат функции Check() (res1). Если $res1 = N$ – ключ найден. Если $res1 = N - 1$, необходимо проинвертировать k-й бит – ключ найден. Конец.
- 7) Переход на п.2.

Примечание. В данном алгоритме абсолютно неважно, какое начальное значение ключа будет использовано, и в каком порядке будут проверяться биты (слева-направо или справа-налево). Важно, что на каждом шаге изменяется ТОЛЬКО один бит.

Ответ: Начиная с любого набора изменять по одному биту. Если результат выполнения функции Check() изменился в БОльшую сторону – угадали значение изменённого бита. Если результат изменился в мЕньшую сторону – НЕ угадали значение измененного бита, необходимо вернуть исходное значение этого бита.