



**МЕЖРЕГИОНАЛЬНАЯ ОЛИМПИАДА ШКОЛЬНИКОВ
ИМЕНИ И.Я. ВЕРЧЕНКО**

**Профиль:
Информатика и компьютерная безопасность**

**Задачи с решениями
(8-10 класс)**

Задача 1. Парольная защита	2
Вариант 1	2
Вариант 2	3
Задача 2. Стеганография	4
Вариант 1	4
Вариант 2	6
Задача 3. Проверка пароля	7
Вариант 1	7
Вариант 2	9
Задача 4. Сетевой вирус	11
Вариант 1	11
Вариант 2	12
Задача 5. Тайное послание	12
Вариант 1	12
Вариант 2	14

Задача 1. Парольная защита

Вариант 1

В соответствии с требованиями службы безопасности Ольга еженедельно меняет парольную фразу для доступа к зашифрованному контейнеру на рабочем компьютере. В качестве парольных фраз она использует фразы из своей любимой книги «Маленький принц» на английском языке. Для того чтобы помнить текущий пароль, Ольга оставляет подсказки на своем рабочем месте.

Посмотрите на фотографию рабочего места Ольги и восстановите текущую парольную фразу. Парольная фраза может состоять из больших и маленьких букв латинского алфавита, цифр, символов и знаков пробела.

К задаче прилагаются:

- 1) файл [The Little Prince.pdf](#),
- 2) фотография рабочего места [Workspace v1.png](#).

Решение

Для определения парольной фразы необходимо найти на рабочем столе подсказки.

На рабочем месте можно заметить цифры в трех местах:

- (1) – номер на баскетбольной майке в рамке на стене,
- (2) – время на рабочем столе компьютера,
- (3) – даты на настольном календаре.



Рисунок 1 – Числа-подсказки

Время на рабочем столе постоянно изменяется, поэтому эти числа не рассматриваем. Остаются два значения:

- «23» на майке,
- «2», выделенная кружочком в календаре.

Поскольку к задаче дан еще и текстовый документ, можно предположить, что парольная фраза спрятана в книжке, а найденные числа - номер страницы и номер строки.

- Два возможных варианта парольной фразы:
- страница 2 строка 23 и страница 23 строка 2.

На странице 2 всего три строки, поэтому этот вариант исключается. Остается страница 23, строка 2, в которой написано: “*Before becoming so big baobabs were small.*” – это и есть парольная фраза.

Ответ: “Before becoming so big baobabs were small.”

Вариант 2

В соответствии с требованиями службы безопасности Ольга еженедельно меняет парольную фразу для доступа к зашифрованному контейнеру на рабочем компьютере. В качестве парольных фраз она использует фразы из своей любимой книги «Маленький принц» на английском языке. Для того чтобы помнить текущий пароль, Ольга оставляет подсказки на своем рабочем месте.

Посмотрите на фотографию рабочего места Ольги и восстановите текущую парольную фразу. Парольная фраза может состоять из больших и маленьких букв латинского алфавита, цифр, символов и знаков пробела.

К задаче прилагаются:

- 1) файл [The Little Prince.pdf](#),
- 2) фотография рабочего места [Workspace v2.png](#).

Решение

Для определения парольной фразы необходимо найти на рабочем столе подсказки. На рабочем месте можно заметить цифры в трех местах:

- (1) – номер на баскетбольной майке в рамке на стене,
- (2) – время на рабочем столе компьютера,
- (3) – даты на настольном календаре.



Рисунок 1 – Числа-подсказки

Время на рабочем столе постоянно изменяется, поэтому эти числа не рассматриваем. Остаются два значения:

«33» на майке,

«15», выделенная кружочком в календаре.

Поскольку к задаче дан еще и текстовый документ, можно предположить, что парольная фраза спрятана в книжке, а найденные числа - номер страницы и номер строки.

Два возможных варианта парольной фразы:

страница 15 строка 33 и страница 33 строка 15.

На странице 15 нет строки 33, поэтому этот вариант исключается. Остается страница 33, строка 15, в которой написано: “*You look so beautiful!*” – это и есть парольная фраза.

Ответ: “You look so beautiful!”

Задача 2. Стеганография

Вариант 1

ВМР (англ. BitMap Picture) – аппаратно-независимое побитовое изображение Windows, используемое для хранения растровых изображений. Дамп памяти изображения размером 5x3 пикселя показан на рисунке ниже.

Offset	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F
00000000	42	4D	66	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	05	00	00	00	03	00	00	00	01	00	18	00	00	00
00000020	00	00	30	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	00	00	00	FF						
00000040	FF	FF	00	00	00	00	FF									
00000050	FF	FF	FF	FF	FF	00	00	00	00	FF						
00000060	FF	FF	00	00	00	00	00	00	00	00	00	00	00	00	00	00

Рисунок – Дамп памяти изображения в формате ВМР

Все байты изображения представлены в шестнадцатеричной системе счисления.

Число 36h, записанное по адресу 000Ah, указывает, с какого адреса начинается запись картинки (это смещение от начала файла, длина заголовка). По адресу 0012h указана ширина рисунка в пикселях. В данном случае число пикселей равно 5. Высота рисунка указывается в ячейке по адресу 0016h (для рассматриваемого рисунка высота – 3 пикселя). В ячейке с адресом 001Ch указана глубина цвета. В данном случае число 18h означает, что для формирования цвета каждого пикселя используется 24 бита (по 8 бит на каждую цветовую составляющую – красную(R), зеленую(G), синюю(B)).

На рисунке ниже структура представлена более детально.



Рисунок – Заголовок ВМР-файла

Особенность формата ВМР заключается в том, что размер каждой пиксельной строки должен быть кратен 4-м байтам. Поэтому помимо самих пикселей в дампе изображения могут встречаться выравнивающие байты, заполняющие длину пиксельной строки до размера, кратного 4.

Известно, что в картинку было внедрено секретное сообщение так, что изображение не было изменено. Найдите скрытое сообщение.

0000014d	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	42	4d	fa	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	09	00	00	00	07	00	00	00	01	00	18	00	00	00
00000020	00	00	c4	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	cc	48	3f	57	7a	b9	57	7a	b9	4c
00000040	b1	22	ea	d9	99	4c	b1	22	1d	e6	b5	27	7f	ff	57	7a
00000050	b9	65	c3	c3	c3	a4	49	a3	cc	48	3f	27	7f	ff	a4	49
00000060	a3	1d	e6	b5	4c	b1	22	cc	48	3f	57	7a	b9	78	cc	48
00000070	3f	c3	c3	c3	c9	ae	ff	4c	b1	22	a4	49	a3	4c	b1	22
00000080	1d	e6	b5	c3	c3	c3	a4	49	a3	61	27	7f	ff	a4	49	a3
00000090	cc	48	3f	27	7f	ff	b0	e4	ef	27	7f	ff	b0	e4	ef	57
000000a0	7a	b9	57	7a	b9	6d	a4	49	a3	57	7a	b9	c3	c3	c3	b0
000000b0	e4	ef	a4	49	a3	b0	e4	ef	c3	c3	c3	a4	49	a3	57	7a
000000c0	b9	70	c9	ae	ff	57	7a	b9	c9	ae	ff	cc	48	3f	a4	49
000000d0	a3	cc	48	3f	c9	ae	ff	a4	49	a3	27	7f	ff	6c	27	7f
000000e0	ff	a4	49	a3	c9	ae	ff	a4	49	a3	27	7f	ff	57	7a	b9
000000f0	cc	48	3f	57	7a	b9	57	7a	b9	65

К задаче прилагается:
изображение [pic_v1.bmp](#).

Решение

Рассмотрим изображение и разберем заголовки формата BMP. В данном изображении ширина составляет 9 пикселей (байт со смещением 0012h), а длина – 7 пикселей (байт со смещением 0016h). На каждый пиксель выделяется $18h = 24_{10}$ бит = 3 байта (ячейка со смещением 001Ch). Соответственно каждая строка занимает $9*3 = 27$ байт + выравнивающий 1 байт (для кратности 4). Общее число таких групп по $27+1$ байт равно 7.

Байты изображения начинаются с ячейки 0036h (ячейка со смещением 00Ah), далее следует $9*3=27$ байтов, описывающих пиксели первой строки изображения. Следовательно, первый выравнивающий байт расположен в ячейке с адресом 0051h. Его значение равно 65h. Аналогично необходимо найти расположение всех 7-ми выравнивающих байтов.

0000014d	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	42	4d	fa	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	09	00	00	00	07	00	00	00	01	00	18	00	00	00
00000020	00	00	c4	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	cc	48	3f	57	7a	b9	57	7a	b9	4c
00000040	b1	22	ea	d9	99	4c	b1	22	1d	e6	b5	27	7f	ff	57	7a
00000050	b9	65	c3	c3	c3	a4	49	a3	cc	48	3f	27	7f	ff	a4	49
00000060	a3	1d	e6	b5	4c	b1	22	cc	48	3f	57	7a	b9	78	cc	48
00000070	3f	c3	c3	c3	c9	ae	ff	4c	b1	22	a4	49	a3	4c	b1	22
00000080	1d	e6	b5	c3	c3	c3	a4	49	a3	61	27	7f	ff	a4	49	a3
00000090	cc	48	3f	27	7f	ff	b0	e4	ef	27	7f	ff	b0	e4	ef	57
000000a0	7a	b9	57	7a	b9	6d	a4	49	a3	57	7a	b9	c3	c3	c3	b0
000000b0	e4	ef	a4	49	a3	b0	e4	ef	c3	c3	c3	a4	49	a3	57	7a
000000c0	b9	70	c9	ae	ff	57	7a	b9	c9	ae	ff	cc	48	3f	a4	49
000000d0	a3	cc	48	3f	c9	ae	ff	a4	49	a3	27	7f	ff	6c	27	7f
000000e0	ff	a4	49	a3	c9	ae	ff	a4	49	a3	27	7f	ff	57	7a	b9
000000f0	cc	48	3f	57	7a	b9	57	7a	b9	65

Рисунок 1 – Выравнивающие байты

В результате получаем следующие байты:

65h 78h 61h 6Dh 70h 6Ch 65h

Используя ASCII-таблицу получим символы:

65h – “e”, 78h – “x”, 61h – “a”, 6Dh – “m”, 70h – “p”, 6Ch – “l”, 65h – “e”.

Секретное сообщение – “example”.

Ответ: “example”

К задаче прилагается:
изображение [pic_v2.bmp](#).

Решение

Рассмотрим изображение и разберем заголовок формата BMP. В данном изображении ширина составляет 9 пикселей (байт со смещением 0012h), а длина – 7 пикселей (байт со смещением 0016h). На каждый пиксель выделяется $18h = 24_{10}$ бит = 3 байта (ячейка со смещением 001Ch). Соответственно каждая строка занимает $9*3 = 27$ байт + выравнивающий 1 байт (для кратности 4). Общее число таких групп по $27+1$ байт равно 7.

Байты изображения начинаются с ячейки 0036h (ячейка со смещением 00Ah), далее следует $9*3=27$ байтов, описывающих пиксели первой строки изображения. Следовательно, первый выравнивающий байт расположен в ячейке с адресом 0051h. Его значение равно 61h. Аналогично необходимо найти расположение всех 7-ми выравнивающих байтов.

0000014d	00	01	02	03	04	05	06	07	08	09	0a	0b	0c	0d	0e	0f
00000000	42	4d	fa	00	00	00	00	00	00	00	36	00	00	00	28	00
00000010	00	00	09	00	00	00	07	00	00	00	01	00	18	00	00	00
00000020	00	00	c4	00	00	00	00	00	00	00	00	00	00	00	00	00
00000030	00	00	00	00	00	00	cc	48	3f	57	7a	b9	57	7a	b9	4c
00000040	b1	22	ea	d9	99	4c	b1	22	1d	e6	b5	27	7f	ff	57	7a
00000050	b9	61	c3	c3	c3	a4	49	a3	cc	48	3f	27	7f	ff	a4	49
00000060	a3	1d	e6	b5	4c	b1	22	cc	48	3f	57	7a	b9	72	cc	48
00000070	3f	c3	c3	c3	c9	ae	ff	4c	b1	22	a4	49	a3	4c	b1	22
00000080	1d	e6	b5	c3	c3	c3	a4	49	a3	74	27	7f	ff	a4	49	a3
00000090	cc	48	3f	27	7f	ff	b0	e4	ef	27	7f	ff	b0	e4	ef	57
000000a0	7a	b9	57	7a	b9	69	a4	49	a3	57	7a	b9	c3	c3	c3	b0
000000b0	e4	ef	a4	49	a3	b0	e4	ef	c3	c3	c3	a4	49	a3	57	7a
000000c0	b9	63	c9	ae	ff	57	7a	b9	c9	ae	ff	cc	48	3f	a4	49
000000d0	a3	cc	48	3f	c9	ae	ff	a4	49	a3	27	7f	ff	6c	27	7f
000000e0	ff	a4	49	a3	c9	ae	ff	a4	49	a3	27	7f	ff	57	7a	b9
000000f0	cc	48	3f	57	7a	b9	57	7a	b9	65

Рисунок 1 – Выравнивающие байты

В результате получаем следующие байты:

61h 72h 74h 69h 63h 6Ch 65h

Используя ASCII-таблицу получим символы:

61h – “a”, 72h – “r”, 74h – “t”, 69h – “i”, 63h – “c”, 6Ch – “l”, 65h – “e”.

Секретное сообщение – “article”.

Ответ: “article”

Задача 3. Проверка пароля

Вариант 1

Ниже представлен листинг кода на языке C++, выполняющий проверку введенного пароля по определенным параметрам. Определите пароль, при котором программа выведет фразу «Password is correct». Ответ обоснуйте.

```
C++
#include <iostream>

using namespace std;
int main()
{
    int password;
    cin >> password;
```

```

if (password<100000 || password>999999)
    cout << "Password is incorrect\n";
else
{
    if (password%39!=0 || password%71!=0)
        cout << "Password is incorrect\n";
    else
    {
        if (password/100000+(password/1000)%10+(password/10)%10!=21 ||
(password/10000)%10+(password/100)%10+password%10!=21)
            cout << "Password is incorrect\n";
        else
        {
            cout << "Password is correct\n";
        }
    }
}
}
}

```

Решение

Пронумеруем строки исходного кода.

1.	#include <iostream>
2.	using namespace std;
3.	int main() {
4.	int password;
5.	cin >> password;
6.	if (password<100000 password>999999)
7.	cout << "Password is incorrect\n";
8.	else {
9.	if (password%39!=0 password%71!=0)
10.	cout << "Password is incorrect\n";
11.	else {
12.	if (password/100000+(password/1000)%10+(password/10)%10!=21 (password/10000)%10+(password/100)%10+password%10!=21)
13.	cout << "Password is incorrect\n";
14.	else {
15.	cout << "Password is correct\n";
16.	} } } }

Способ 1

Написать программу перебора всех комбинаций паролей и поиск по условиям:

- 1) Из условия (6) следует вывод, что пароль - число в диапазоне от 100000 до 999999 включительно.
- 2) Условие (9) показывает, что пароль должен делиться нацело на 39 и 71.
- 3) Условие (12) проверяет, что сумма 1й, 3й и 5й цифр пароля и сумма 2й, 4й, 6й цифр пароля должны равняться 21.
- 4) После прохождения всех проверок вывести число на экран.

Примерный код перебора всех возможных комбинаций приведен в листинге.

Листинг 1. Перебор паролей

```
#include <iostream>

using namespace std;

int main()
{
    for (int i=100000; i<=999999; i++)
    {
        if (i%39==0 && i%71==0)
        {
            int summ1=i/100000+(i/1000)%10+(i/10)%10;
            int summ2=(i/10000)%10+(i/100)%10+i%10;
            if (summ1==summ2 && summ1==21)
            {
                cout<<i<<endl;
            }
        }
    }
}
```

Способ 2

Подобрать пароль вручную. Условия на пароль:

- 1) 6-значная комбинация, состоящая из цифр,
- 2) пароль должен быть кратным 39 и 71,
- 3) сумма цифр на нечетных позициях должна равняться сумме цифр на четных позициях и быть равной 21.

Таких комбинаций всего одна – 974688.

Ответ: 974688

Вариант 2

Ниже представлен листинг кода на языке C++, выполняющий проверку введенного пароля по определенным параметрам. Определите пароль, при котором программа выведет фразу «*Password is correct*». Ответ обоснуйте.

<pre>C++ #include <iostream> using namespace std; int main() { int password; cin>>password; if (password<100000 password>999999) cout<<"Password is incorrect\n"; else { if (password%29!=0 password%43!=0) cout<<"Password is incorrect\n"; else { if (password/100000+(password/1000)%10+(password/10)%10!=9 (password/10000)%10+(password/100)%10+password%10!=9) cout<<"Password is incorrect\n"; } } }</pre>

```

        else
        {
            cout<<"Password is correct\n";
        }
    }
}

```

Решение

Пронумеруем строки исходного кода.

1.	#include <iostream>
2.	using namespace std;
3.	int main() {
4.	int password;
5.	cin >> password;
6.	if (password<100000 password>999999)
7.	cout << "Password is incorrect\n";
8.	else {
9.	if (password%29!=0 password%43!=0)
10.	cout << "Password is incorrect\n";
11.	else {
12.	if (password/100000+(password/1000)%10+(password/10)%10!=9 (password/10000)%10+(password/100)%10+password%10!=9)
13.	cout << "Password is incorrect\n";
14.	else {
15.	cout << "Password is correct\n";
16.	} } } }

Способ 1

Написать программу перебора всех комбинаций паролей и поиск по условиям:

- 1) Из условия (6) следует вывод, что пароль - число в диапазоне от 100000 до 999999 включительно.
- 2) Условие (9) показывает, что пароль должен делиться нацело на 29 и 43.
- 3) Условие (12) проверяет, что сумма 1й, 3й и 5й цифр пароля и сумма 2й, 4й, 6й цифр пароля должны равняться 9.
- 4) После прохождения всех проверок вывести число на экран.

Примерный код перебора всех возможных комбинаций приведен в листинге.

Листинг 1. Перебор паролей

```

#include <iostream>

using namespace std;

int main()
{
    for (int i=100000; i<=999999; i++)
    {
        if (i%29==0 && i%43==0)
        {

```

```

int summ1=i/100000+(i/1000)%10+(i/10)%10;
int summ2=(i/10000)%10+(i/100)%10+i%10;
if (summ1==summ2 && summ1==9)
{
    cout<<i<<endl;
}
}
}
}

```

Способ 2

Подобрать пароль вручную. Условия на пароль:

- 1) 6-значная комбинация, состоящая из цифр,
- 2) пароль должен быть кратным 29 и 43,
- 3) сумма цифр на нечетных позициях должна равняться сумме цифр на четных позициях и быть равной 9.

Таких комбинаций всего одна – 123453.

Ответ: 123453

Задача 4. Сетевой вирус

Вариант 1

Злоумышленник разработал сетевой вредоносный код, который осуществляет отправку пользовательской информации с зараженного компьютера на центральный сервер. Для того, чтобы его сетевая активность не была обнаружена антивирусными программами, создается скрытый канал передачи информации с использованием поля ID (Идентификатор пакета) в заголовке IP-пакета.

4 бита Номер версии	4 бита Длина заголовка	8 бит Тип сервиса				16 бит Общая длина			
		PR	D	T	R				
16 бит Идентификатор пакета						3 бита Флаги		16 бит Контрольная сумма	
		D		M					
8 бит Время жизни		8 бит Протокол верхнего уровня				16 бит Контрольная сумма			
32 бита IP-адрес источника									
32 бита IP-адрес назначения									
Опции и выравнивание									

Рисунок – Заголовок IP-пакета

Размер IP-пакета составляет 1500 байт, в котором заголовок составляет 54 байта, остальное – пользовательские данные.

Файлы какого суммарного объема пользователю зараженного компьютера необходимо отправить в сеть, чтобы вредоносная программа смогла загрузить на центральный сервер 1 Кб пользовательской информации? Ответ обоснуйте.

Решение

- 1) Поле ID в заголовке IPv4 имеет размер 16 бит. Поэтому каждый пакет передает 2 байта информации.
- 2) 1 Кб пользовательской информации разбивается на 512 пакетов (в каждом по 2 байта).

- 3) В каждом пакете передается $1500 - 54 = 1446$ байтов данных.
- 4) Для передачи 512 пакетов требуется $1446 * 512 = 741798$ байт данных.

Ответ: 741798 байт.

Вариант 2

Злоумышленник разработал сетевой вредоносный код, который осуществляет отправку пользовательской информации с зараженного компьютера на центральный сервер. Для того, чтобы его сетевая активность не была обнаружена антивирусными программами, создается скрытый канал передачи информации с использованием поля ID (Идентификатор пакета) в заголовке IP-пакета.

4 бита Номер версии	4 бита Длина заголовка	8 бит Тип сервиса				16 бит Общая длина			
		PR	D	T	R				
16 бит Идентификатор пакета						3 бита Флаги		16 бит Контрольная сумма	
						D	M		
8 бит Время жизни		8 бит Протокол верхнего уровня				16 бит Контрольная сумма			
32 бита IP-адрес источника									
32 бита IP-адрес назначения									
Опции и выравнивание									

Рисунок – Заголовок IP-пакета

Размер IP-пакета составляет 1500 байт, в котором заголовок составляет 54 байта, остальное – пользовательские данные.

Файлы какого суммарного объема пользователю зараженного компьютера необходимо отправить в сеть, чтобы вредоносная программа смогла загрузить на центральный сервер 2 Кб пользовательской информации? Ответ обоснуйте.

Решение

- 1) Поле ID в заголовке IPv4 имеет размер 16 бит. Поэтому каждый пакет передает 2 байта информации.
- 2) 2 Кб пользовательской информации разбивается на 1024 пакетов (в каждом по 2 байта).
- 3) В каждом пакете передается $1500 - 54 = 1446$ байтов данных.
- 4) Для передачи 1024 пакетов требуется $1446 * 1024 = 1480704$ байт данных.

Ответ: 1480704 байт

Задача 5. Тайное послание

Вариант 1

Участники киберсети обмениваются между собой сообщениями с использованием «японского кроссворда».

Каждое число в таком кроссворде напротив строки или столбца обозначает один горизонтальный или вертикальный блок, состоящий из указанного числа подряд идущих закрашенных клеток. Между закрашенными блоками должно быть не менее одной пустой клетки. Количество чисел в строке или столбце определяет количество таких блоков в строке или столбце соответственно.

Например, фрагмент вида:

		2	1	2
1				
1 1				
3				

закрашивается следующим образом:

		2	1	2
1				
1 1				
3				

Аналитик обнаружил очередное зашифрованное сообщение. Помогите аналитику расшифровать его.

				1	1	1	4
			2	1	1	1	1
	0	8	8	2	1	1	1
3 2							
2 1 1							
2 2							
3 1 1							
3 1							
2 1 1							
3 1							
4 1							

Решение

Если закрасить кроссворд в соответствии с условиями, то можно получить следующий результат:

				1	1	1	4
			2	1	1	1	1
	0	8	8	2	1	1	1
3 2							
2 1 1							
2 2							
3 1 1							
3 1							
2 1 1							
3 1							
4 1							

Предположим, что сообщение шифруется с помощью закрашенных и незакрашенных ячеек кроссворда. Пусть незакрашенные ячейки будут обозначать 0, а закрашенные 1.

Тогда кроссворд можно переписать, как

```

01110011
01100101
01100011
01110101
01110010
01101001
01110100
01111001

```

Если перевести полученные значения в десятичный вид и найти соответствующие им символы в ASCII-таблице, то можно получить зашифрованное слово: “security”.

Ответ: “security”

Вариант 2

Участники киберсети обмениваются между собой сообщениями с использованием «японского кроссворда».

Каждое число в таком кроссворде напротив строки или столбца обозначает один горизонтальный или вертикальный блок, состоящий из указанного числа подряд идущих закрашенных клеток. Между закрашенными блоками должно быть не менее одной пустой клетки. Количество чисел в строке или столбце определяет количество таких блоков в строке или столбце соответственно.

Например, фрагмент вида:

		2	1	2
1				
1 1				
3				

закрашивается следующим образом:

		2	1	2
1				
1 1				
3				

Аналитик обнаружил очередное зашифрованное сообщение. Помогите аналитику расшифровать его.

	0	8	8	1 3 1	1	2 1	5	5
3								
2 1								
3 2								
3 2								
3 3								
2 4								
3 1								
2 1								

Решение

Если закрасить кроссворд в соответствии с условиями, то можно получить следующий результат:

	0	8	8	1 3 1	1	2 1	5	5
3								
2 1								
3 2								
3 2								
3 3								
2 4								
3 1								
2 1								

Предположим, что сообщение шифруется с помощью закрашенных и незакрашенных ячеек кроссворда. Пусть незакрашенные ячейки будут обозначать 0, а закрашенные 1.

Тогда кроссворд можно переписать, как

```

01110000
01100001
01110011
01110011
01110111
01101111
01110010
01100100

```

Если перевести полученные значения в десятичный вид и найти соответствующие им символы в ASCII-таблице, то можно получить зашифрованное слово: "password".

Ответ: "password"