

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ
ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ
«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 7

Выполнили:
Студенты группы: ИП-715
Кузнецов Егор и Винтер Антон

Проверил: доцент кафедры ПМиК
Приставка П.А.

№ 1

В скрипте **z07-1.php**

- 1.Создайте массив \$treug[] "треугольных" чисел, т.е. чисел вида $n(n+1)/2$ (где $n=1,2,\dots, 10$) и выведите значения этого массива на экран в строку (через 2 пробела).
- 2.Создайте массив \$kvd[] квадратов натуральных чисел от 1 до 10, выведите значения этого массива на экран в строку.
- 3.Объедините эти 2 массива в массив \$rez[], выведите результат на экран.
- 4.Отсортируйте массив \$rez[], выведите результат на экран.
- 5.Удалите в массиве \$rez[] первый элемент, выведите результат на экран.
- 6.С помощью функции array_unique() удалите из массива \$rez[] повторяющиеся элементы, результат занесите в массив \$rez1[] и выведите его на экран.

(Использовать листинги 9-2 - 9-10).

<?php

```
function printArr($arr) {  
    foreach ($arr as $el) {  
        print "$el ";  
        print "&nbsp;";  
    }  
    print "<br>"; }
```

/////1

```
$treug = array();  
for ($n=1; $n <= 10; $n++) {  
    array_push($treug, $n*($n+1)/2); }  
    print "1) ";  
    printArr($treug);
```

/////2

```
$kvd = array();
```

```
for ($n=1; $n <= 10; $n++) {  
    array_push($skvd, $n*$n); }  
    print "2) ";  
printArr($skvd);
```

/////3

```
$rez = array_merge($streug, $skvd);  
print "3) ";  
printArr($rez);
```

/////4

```
sort($rez);  
print "4) ";  
printArr($rez);
```

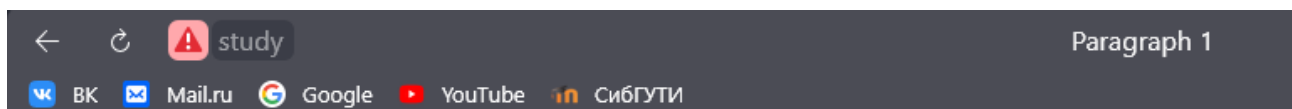
/////5

```
$rm = array_shift($rez);  
print "5) ";  
printArr($rez);
```

/////6

```
$rez1 = array_unique($rez);  
print "6) ";  
printArr($rez1);
```

?>



```
1) 1 3 6 10 15 21 28 36 45 55  
2) 1 4 9 16 25 36 49 64 81 100  
3) 1 3 6 10 15 21 28 36 45 55 1 4 9 16 25 36 49 64 81 100  
4) 1 1 3 4 6 9 10 15 16 21 25 28 36 36 45 49 55 64 81 100  
5) 1 3 4 6 9 10 15 16 21 25 28 36 36 45 49 55 64 81 100  
6) 1 3 4 6 9 10 15 16 21 25 28 36 45 49 55 64 81 100
```

№2

В скрипте z07-2.php

1. Создайте массив `$treug[]` "треугольных" чисел (для n от 1 до 30) и массив квадратов `$kvd[]` (для n от 1 до 30).
2. Используя вложенные циклы `for` отобразите на экране таблицу Пифагора 30×30 (размер чисел в ячейках: `size=1`). В этой таблице фон у ячеек с квадратами чисел должен быть синим, а у ячеек с "треугольными" числами - зеленым. У ячеек, в которых стоят числа, одновременно являющиеся и квадратами и "треугольными" (здесь это числа 1 и 36) фон должен быть красным. У остальных ячеек фон белый. Для проверки правильности закрашивания ячеек, под таблицей выведите "треугольные" числа в строку. В результате должно получиться так:

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
2	4	5	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100	104	108	112	116	120
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126	132	138	144	150	156	162	168	174	180
7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147	154	161	168	175	182	189	196	203	210
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240
9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300
11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209	220	231	242	253	264	275	286	297	308	319	330
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240	252	264	276	288	300	312	324	336	348	360
13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260	273	286	299	312	325	338	351	364	377	390
14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266	280	294	308	322	336	350	364	378	392	406	420
15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390	405	420	435	450
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304	320	336	352	368	384	400	416	432	448	464	480
17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323	340	357	374	391	408	425	442	459	476	493	510
18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342	360	378	396	414	432	450	468	486	504	522	540
19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361	380	399	418	437	456	475	494	513	532	551	570
20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400	420	440	460	480	500	520	540	560	580	600
21	42	63	84	105	126	147	168	189	210	231	252	273	294	315	336	357	378	399	420	441	462	483	504	525	546	567	588	609	630
22	44	66	88	110	132	154	176	198	220	242	264	286	308	330	352	374	396	418	440	462	484	506	528	550	572	594	616	638	660
23	46	69	92	115	138	161	184	207	230	253	276	299	322	345	368	391	414	437	460	483	506	529	552	575	598	621	644	667	690
24	48	72	96	120	144	168	192	216	240	264	288	312	336	360	384	408	432	456	480	504	528	552	576	600	624	648	672	696	720
25	50	75	100	125	150	175	200	225	250	275	300	325	350	375	400	425	450	475	500	525	550	575	600	625	650	675	700	725	750
26	52	78	104	130	156	182	208	234	260	286	312	338	364	390	416	442	468	494	520	546	572	598	624	650	676	702	728	754	780
27	54	81	108	135	162	189	216	243	270	297	324	351	378	405	432	459	486	513	540	567	594	621	648	675	702	729	756	783	810
28	56	84	112	140	168	196	224	252	280	308	336	364	392	420	448	476	504	532	560	588	616	644	672	700	728	756	784	812	840
29	58	87	116	145	174	203	232	261	290	319	348	377	406	435	464	493	522	551	580	609	638	667	696	725	754	783	812	841	870
30	60	90	120	150	180	210	240	270	300	330	360	390	420	450	480	510	540	570	600	630	660	690	720	750	780	810	840	870	900

```
<?php
```

```
    $streug = array();  
    for ($n=1; $n <= 30; $n++) {  
        array_push($streug, $n*($n+1)/2);  
    }
```

```
    $kvd = array();  
    for ($n=1; $n <= 30; $n++) {  
        array_push($kvd, $n*$n);  
    }
```

```
    $color = "silver";  
    print "<table border = 1, cellpadding = 1>";  
    for ($y=1; $y <= 30; $y++) {  
        print "<tr>\n";  
        for ($x=1; $x <= 30; $x++) {  
            $val = $x*$y;  
            $color = "white";  
  
            if (in_array($val, $kvd) and in_array($val, $streug)) {  
                $color = "red";  
            }  
            else if (in_array($val, $streug)) {  
                $color = "green";  
            }  
            else if (in_array($val, $kvd)) {  
                $color = "blue";  
            }  
        }  
    }
```

```

    }

    print "\t<td bgcolor=$color>";

    print ($val);

    print "</td>\n";

}

print "</tr>\n";

}

print "</table>";

```

```

foreach ($streug as $num) {

    print "$num ";

}

print "<br>";

?>

```

← ↻ ⚠ study Paragraph 2

📧 BK 📧 Mail.ru 📍 Google 📺 YouTube 📶 СибГУТИ

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30
2	4	5	8	10	12	14	16	18	20	22	24	26	28	30	32	34	36	38	40	42	44	46	48	50	52	54	56	58	60
3	6	9	12	15	18	21	24	27	30	33	36	39	42	45	48	51	54	57	60	63	66	69	72	75	78	81	84	87	90
4	8	12	16	20	24	28	32	36	40	44	48	52	56	60	64	68	72	76	80	84	88	92	96	100	104	108	112	116	120
5	10	15	20	25	30	35	40	45	50	55	60	65	70	75	80	85	90	95	100	105	110	115	120	125	130	135	140	145	150
6	12	18	24	30	36	42	48	54	60	66	72	78	84	90	96	102	108	114	120	126	132	138	144	150	156	162	168	174	180
7	14	21	28	35	42	49	56	63	70	77	84	91	98	105	112	119	126	133	140	147	154	161	168	175	182	189	196	203	210
8	16	24	32	40	48	56	64	72	80	88	96	104	112	120	128	136	144	152	160	168	176	184	192	200	208	216	224	232	240
9	18	27	36	45	54	63	72	81	90	99	108	117	126	135	144	153	162	171	180	189	198	207	216	225	234	243	252	261	270
10	20	30	40	50	60	70	80	90	100	110	120	130	140	150	160	170	180	190	200	210	220	230	240	250	260	270	280	290	300
11	22	33	44	55	66	77	88	99	110	121	132	143	154	165	176	187	198	209	220	231	242	253	264	275	286	297	308	319	330
12	24	36	48	60	72	84	96	108	120	132	144	156	168	180	192	204	216	228	240	252	264	276	288	300	312	324	336	348	360
13	26	39	52	65	78	91	104	117	130	143	156	169	182	195	208	221	234	247	260	273	286	299	312	325	338	351	364	377	390
14	28	42	56	70	84	98	112	126	140	154	168	182	196	210	224	238	252	266	280	294	308	322	336	350	364	378	392	406	420
15	30	45	60	75	90	105	120	135	150	165	180	195	210	225	240	255	270	285	300	315	330	345	360	375	390	405	420	435	450
16	32	48	64	80	96	112	128	144	160	176	192	208	224	240	256	272	288	304	320	336	352	368	384	400	416	432	448	464	480
17	34	51	68	85	102	119	136	153	170	187	204	221	238	255	272	289	306	323	340	357	374	391	408	425	442	459	476	493	510
18	36	54	72	90	108	126	144	162	180	198	216	234	252	270	288	306	324	342	360	378	396	414	432	450	468	486	504	522	540
19	38	57	76	95	114	133	152	171	190	209	228	247	266	285	304	323	342	361	380	399	418	437	456	475	494	513	532	551	570
20	40	60	80	100	120	140	160	180	200	220	240	260	280	300	320	340	360	380	400	420	440	460	480	500	520	540	560	580	600
21	42	63	84	105	126	147	168	189	210	231	252	273	294	315	336	357	378	399	420	441	462	483	504	525	546	567	588	609	630
22	44	66	88	110	132	154	176	198	220	242	264	286	308	330	352	374	396	418	440	462	484	506	528	550	572	594	616	638	660
23	46	69	92	115	138	161	184	207	230	253	276	299	322	345	368	391	414	437	460	483	506	529	552	575	598	621	644	667	690
24	48	72	96	120	144	168	192	216	240	264	288	312	336	360	384	408	432	456	480	504	528	552	576	600	624	648	672	696	720
25	50	75	100	125	150	175	200	225	250	275	300	325	350	375	400	425	450	475	500	525	550	575	600	625	650	675	700	725	750
26	52	78	104	130	156	182	208	234	260	286	312	338	364	390	416	442	468	494	520	546	572	598	624	650	676	702	728	754	780
27	54	81	108	135	162	189	216	243	270	297	324	351	378	405	432	459	486	513	540	567	594	621	648	675	702	729	756	783	810
28	56	84	112	140	168	196	224	252	280	308	336	364	392	420	448	476	504	532	560	588	616	644	672	700	728	756	784	812	840
29	58	87	116	145	174	203	232	261	290	319	348	377	406	435	464	493	522	551	580	609	638	667	696	725	754	783	812	841	870
30	60	90	120	150	180	210	240	270	300	330	360	390	420	450	480	510	540	570	600	630	660	690	720	750	780	810	840	870	900

1 3 6 10 15 21 28 36 45 55 66 78 91 105 120 136 153 171 190 210 231 253 276 300 325 351 378 406 435 465

№3

С помощью скрипта **z07-3.php** отобразите на экране таблицу Пифагора 30×30 (border=1, отступ содержимого ячеек от границы равен 0, ширина ячейки 14 пикселей, высота ячейки 15 пикселей, размер символов в ячейке size=1, но вместо чисел поставьте неразрывный пробел:).

Фон ячеек определяется в зависимости от того, чему равен остаток от деления числа в ячейке на 7 следующим образом:

если остаток равен 0, то фон белый (white),

если 1 - голубой (aqua),

если 2 - синий (blue),

если 3 - желтый (yellow),

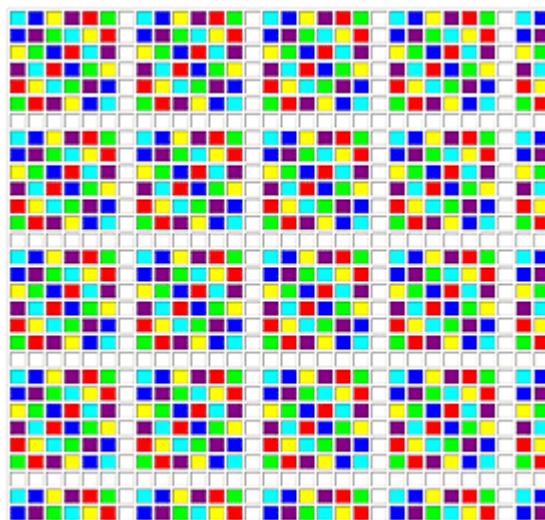
если 4 - фиолетовый (purple),

если 5 - красный (red)

и если 6 - лимонный (lime)

(здесь можно использовать либо оператор if, либо switch).

Посмотрите на получившийся узор:



```
<?php
```

```
    print "<table border = 1 cellpadding = 0>";
```

```
    for ($y=1; $y <= 30; $y++) {
```

```
        print "<tr>\n";
```

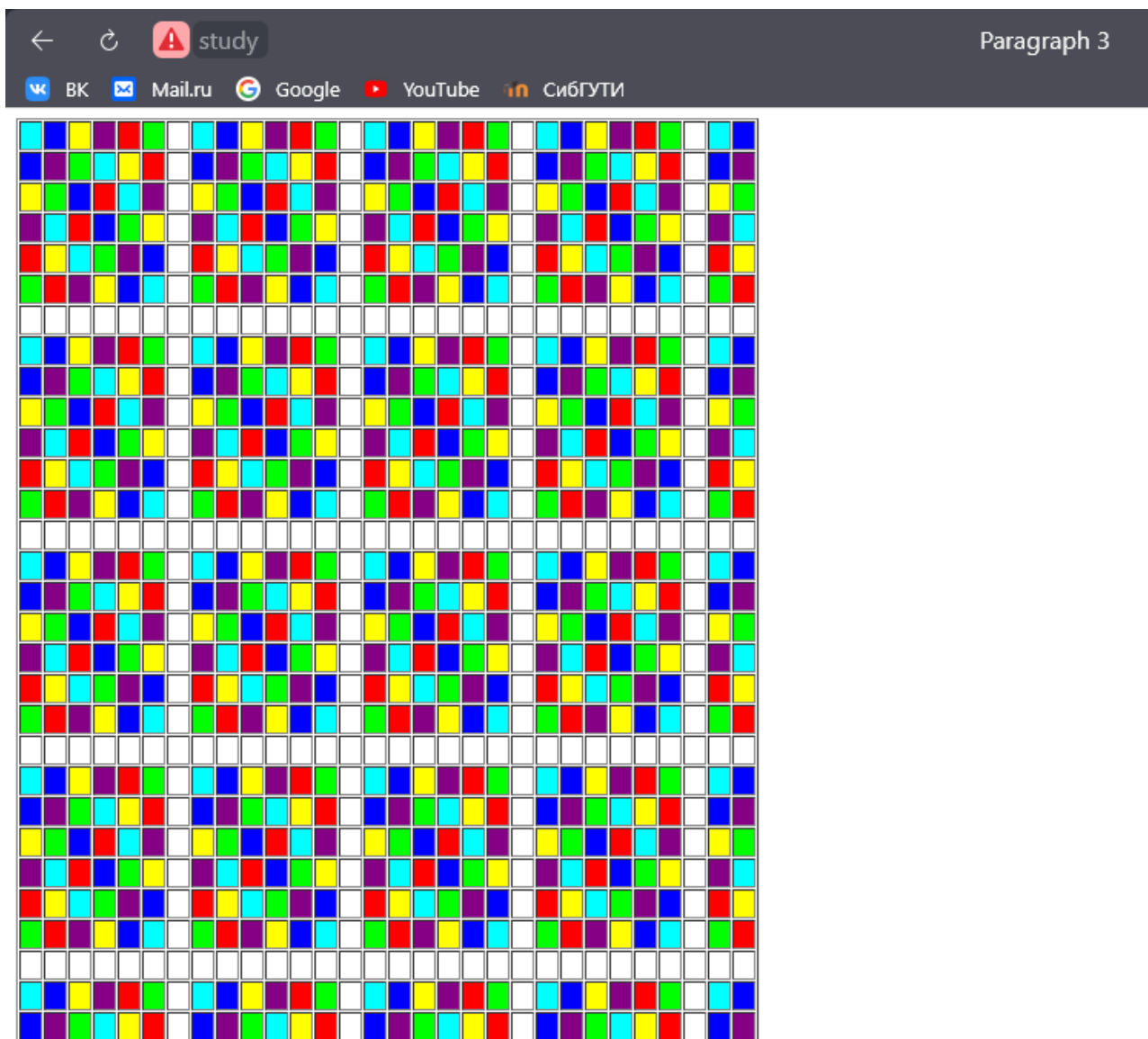
```
for ($x=1; $x <= 30; $x++) {  
    $val = $x*$y;  
    $left = $val % 7;  
  
    switch ($left) {  
        case 1:  
            $color = "aqua";  
            break;  
        case 2:  
            $color = "blue";  
            break;  
        case 3:  
            $color = "yellow";  
            break;  
        case 4:  
            $color = "purple";  
            break;  
        case 5:  
            $color = "red";  
            break;  
        case 6:  
            $color = "lime";  
            break;  
        default:  
            $color = "white";  
    }  
  
    print "\t<td width='14' height='15' bgcolor=$color>";
```



```

        print ("&nbsp;");
        print "</td>\n";
    }
    print "</tr>\n";
}
print "</table>";
?>

```

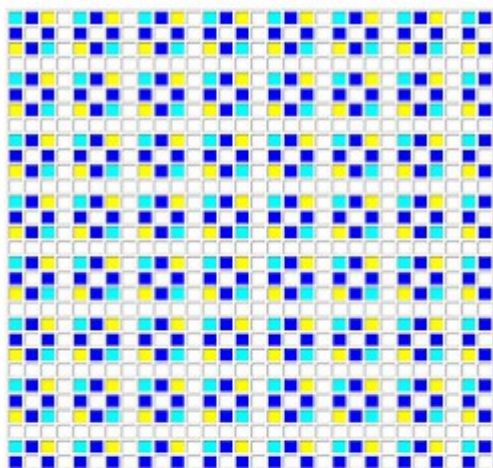


№4

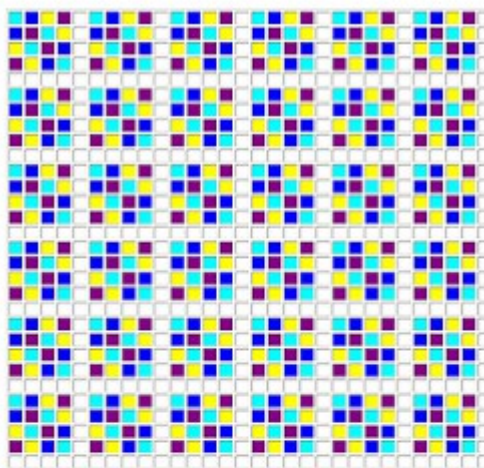
В скрипте z07-4.php

1. Создайте массив `$colors[]`, элементами которого являются цвета: белый, голубой, синий, желтый, фиолетовый, красный и лимонный (т.е. последовательность цветов из предыдущего скрипта).
2. Используя таблицу Пифагора из предыдущего скрипта (z07-3.php), добавьте еще один внешний цикл, его индекс k - это число (от 4 до 7), в зависимости от остатков от деления на которое ячейки будут раскрашиваться в цвета из массива `$colors[]`. Таким образом, должно получиться 4 таблицы-мозаики. Перед каждой таблицей выведите чему равно k (т.е. на что мы делим).

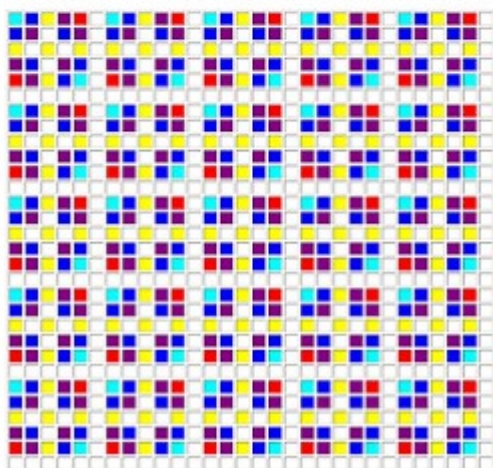
$k=4$



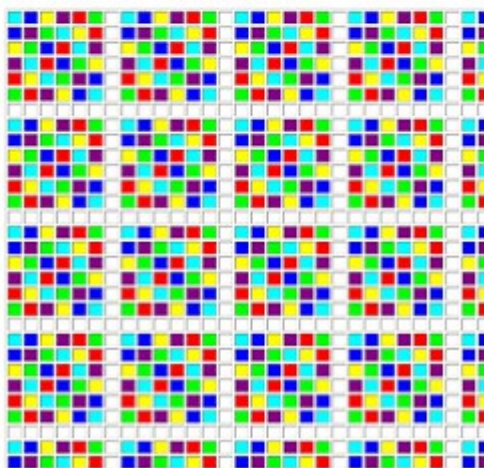
$k=5$



$k=6$



$k=7$



```
<?php
```

```
    $colors = array("white", "aqua", "blue", "yellow", "purple", "red", "lime");
```

```
for ($k=4; $k <= 7; $k++){
```

```
    print "<br>";
```

```
    print "k = $k\n";
```

```
    print "<table cellpadding='0' border='1'>\n";
```

```
for ($y=1; $y <= 30; $y++) {
```

```
    print "<tr>";
```

```
    for ($x=1; $x <= 30; $x++) {
```

```
        $val = $x*$y;
```

```
        $left = $val % $k;
```

```
    switch ($left) {
```

```
        case 1:
```

```
            $color = "aqua";
```

```
            break;
```

```
        case 2:
```

```
            $color = "blue";
```

```
            break;
```

```
        case 3:
```

```
            $color = "yellow";
```

```
            break;
```

```
        case 4:
```

```
            $color = "purple";
```

```
            break;
```

```
        case 5:
```

```
    $color = "red";  
    break;  
case 6:  
    $color = "lime";  
    break;  
default:  
    $color = "white";  
}
```

```
print "\t<td width='14' height='15' bgcolor=$color>";  
print ("&nbsp;");  
print "</td>\n";  
}
```

```
print "</tr>\n";
```

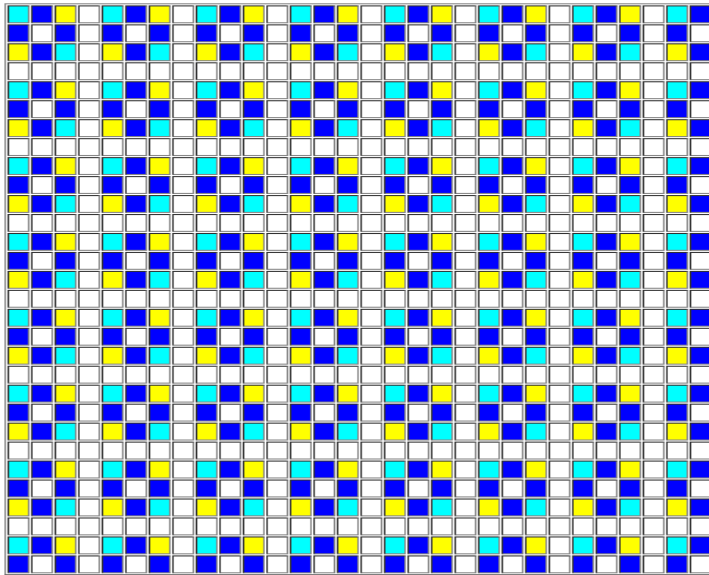
```
}
```

```
print "</table>";
```

```
}
```

```
?>
```

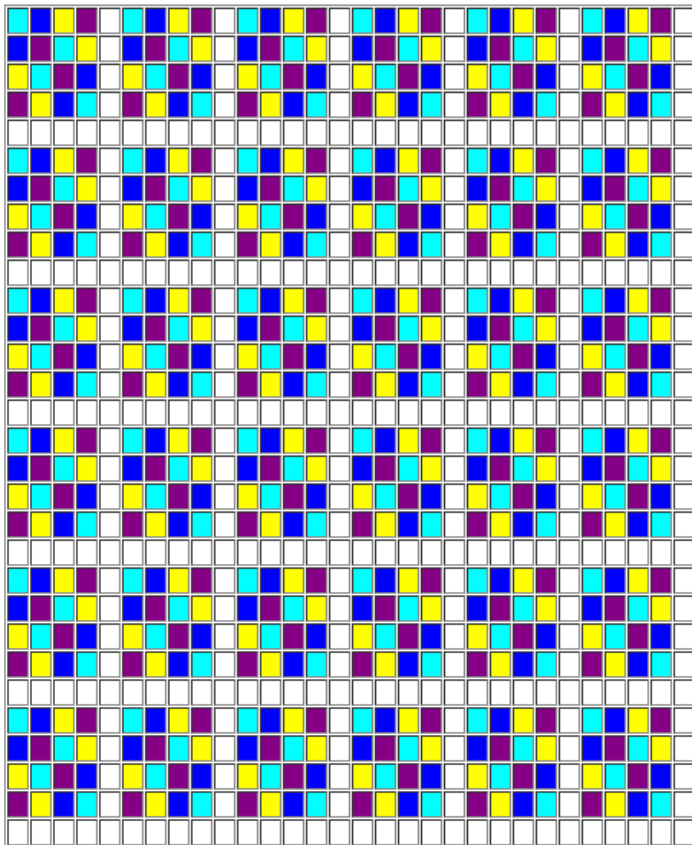
k = 4



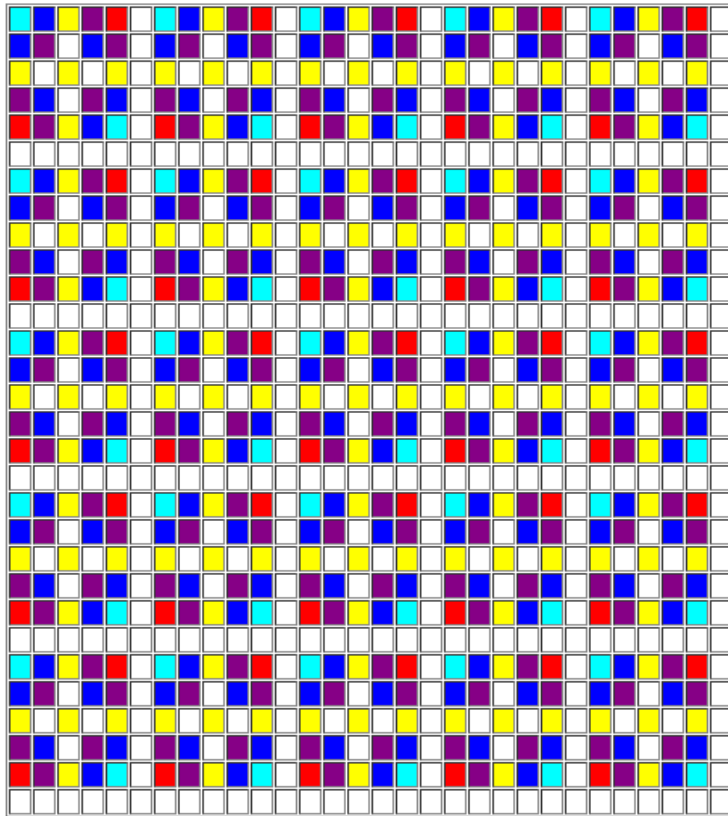
k = 5



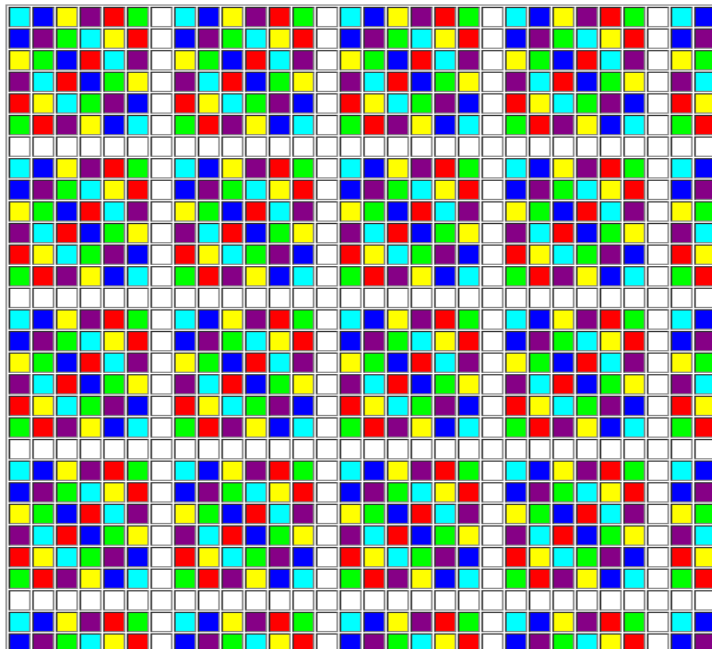
k = 5



k = 6



k = 7



№5

В скрипте **z07-5.php**

1. Создайте ассоциативный массив `$cust[]` с ключами `cnum`, `cname`, `city`, `snum` и `rating` и значениями: 2001, Hoffman, London, 1001 и 100.

Выведите этот массив (вместе с именами ключей) на экран.

2. Отсортируйте этот массив по значениям. Выведите результат на экран.

3. Отсортируйте этот массив по ключам. Выведите результат на экран.

4. Выполните сортировку массива с помощью функции `sort()`. Выведите результат на экран и объясните что получилось.

(Использовать листинги 9-1 - 9-10).

```
<?php
```

```
//////1
```

```
    $cust = array (
```

```
        'cnum' => 2001,
```

```
        'cname' => "Hoffman",
```

```
        'city' => "London",
```

```
        'snum' => 1001,
```

```
        'rating' => 100
```

```
    );
```

```
        print "1)<br>";
```

```
foreach($cust as $c => $c_value) {
```

```
    echo "Key=" . $c . ", Value=" . $c_value;
```

```
    echo "<br>";
```

```
}
```

/////2

```
asort($cust);  
    print "<br> 2)<br>";  
  
foreach($cust as $c => $c_value) {  
    echo "Key=" . $c . ", Value=" . $c_value;  
    echo "<br>";  
}
```

/////3

```
ksort($cust);  
print("<br> 3) <br>");  
  
foreach($cust as $c => $c_value) {  
    echo "Key=" . $c . ", Value=" . $c_value;  
    echo "<br>";  
}
```

/////4

```
sort($cust);  
print("<br> 4) <br>");  
  
foreach($cust as $c => $c_value) {  
    echo "Key=" . $c . ", Value=" . $c_value;  
    echo "<br>";  
}
```

/*asort(\$cust); //пример сортировки ассоциативного массива через asort()

```
print("<br> 4) <br>");  
foreach($cust as $c => $c_value) {
```



```
        echo "Key=" . $c . ", Value=" . $c_value;  
    echo "<br>";  
}*/  
?>
```



```
1)  
Key=cnum, Value=2001  
Key=cname, Value=Hoffman  
Key=city, Value=London  
Key=snum, Value=1001  
Key=rating, Value=100  
  
2)  
Key=cname, Value=Hoffman  
Key=city, Value=London  
Key=rating, Value=100  
Key=snum, Value=1001  
Key=cnum, Value=2001  
  
3)  
Key=city, Value=London  
Key=cname, Value=Hoffman  
Key=cnum, Value=2001  
Key=rating, Value=100  
Key=snum, Value=1001  
  
4)  
Key=0, Value=Hoffman  
Key=1, Value=London  
Key=2, Value=100  
Key=3, Value=1001  
Key=4, Value=2001
```

Т.к. мы передали функции `sort()` ассоциативный массив, то в нашем массиве вместо имен будут сохранены упорядоченные числа.

Вместо функции `sort()` можно использовать `asort()` или `ksort()`, смотря по какому параметру хотим отсортировать.