ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ

ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ТЕЛЕКОММУНИКАЦИЙ И ИНФОРМАТИКИ»

ЛАБОРАТОРНАЯ РАБОТА № 7

Выполнил:

Студент группы: ИП-712

Алексеев С.В.

Проверил: профессор кафедры ПМиК Барахнин В.Б.

Новосибирск – 2020

# № 1

В скрипте **z07-1.php**

1.Создайте массив $treug[] "треугольных" чисел, т.е. чисел вида n(n+1)/2 (где n=1,2,… 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; $n++) { array\_push($kvd, $n\*$n); } print "2) ";

printArr($kvd);

//////3

$rez = array\_merge($treug, $kvd); print "3) "; printArr($rez);

//////4 sort($rez); print "4) ";

printArr($rez);

//////5

$rm = array\_shift($rez); print "5) ";

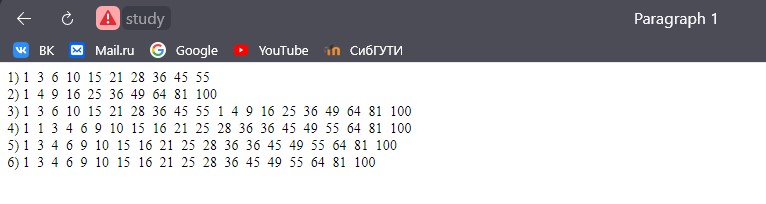
printArr($rez);

//////6

$rez1 = array\_unique($rez); print "6) ";

printArr($rez1);

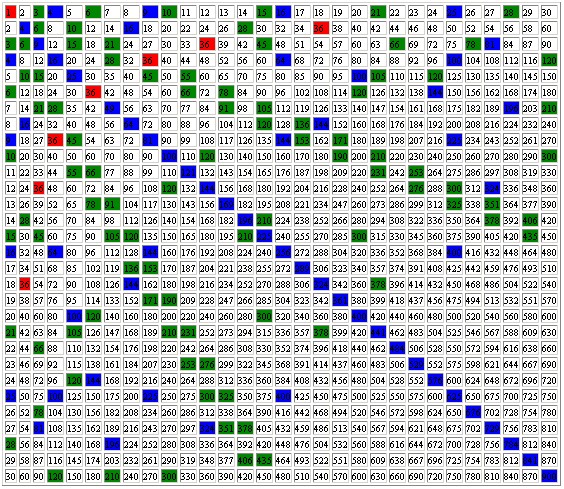
?>



# №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:



<?php

$treug = array(); for ($n=1; $n <= 30; $n++) { array\_push($treug, $n\*($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, $treug)) {

$color = "red";

}

else if (in\_array($val, $treug)) {

$color = "green";

}

else if (in\_array($val, $kvd)) {

$color = "blue";

}

print "\t<td bgcolor=$color>"; print ($val); print "</td>\n";

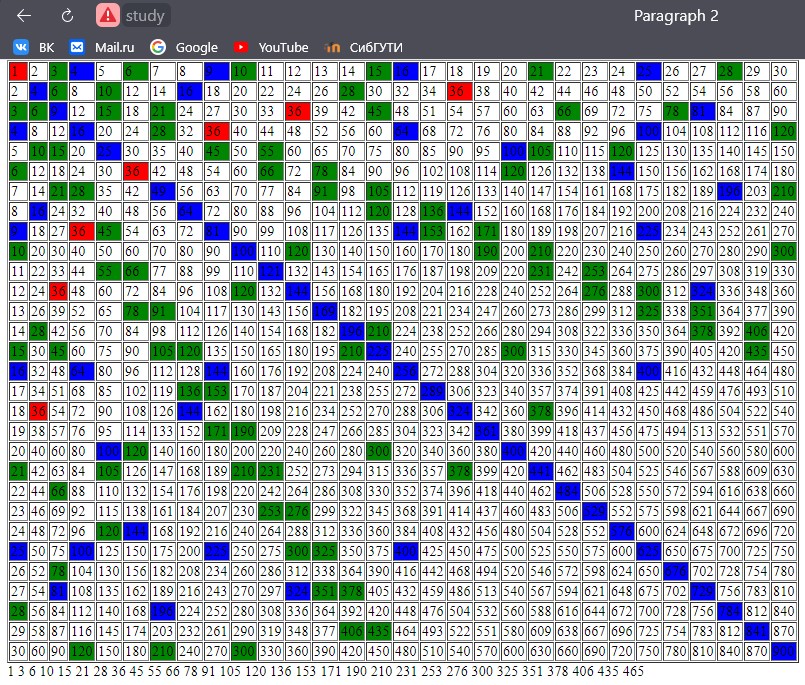
} print "</tr>\n"; } print "</table>";

foreach ($treug as $num) { print "$num ";

}

print "<br>";

?>



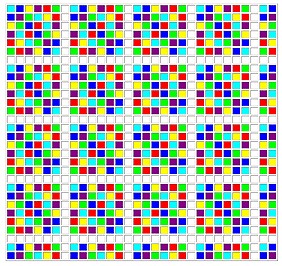
# №3

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 пикселов, высота ячейки 15 пикселов, размер символов в ячейке size=1, но вместо чисел поставьте неразрывный пробел: &nbsp;).

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:

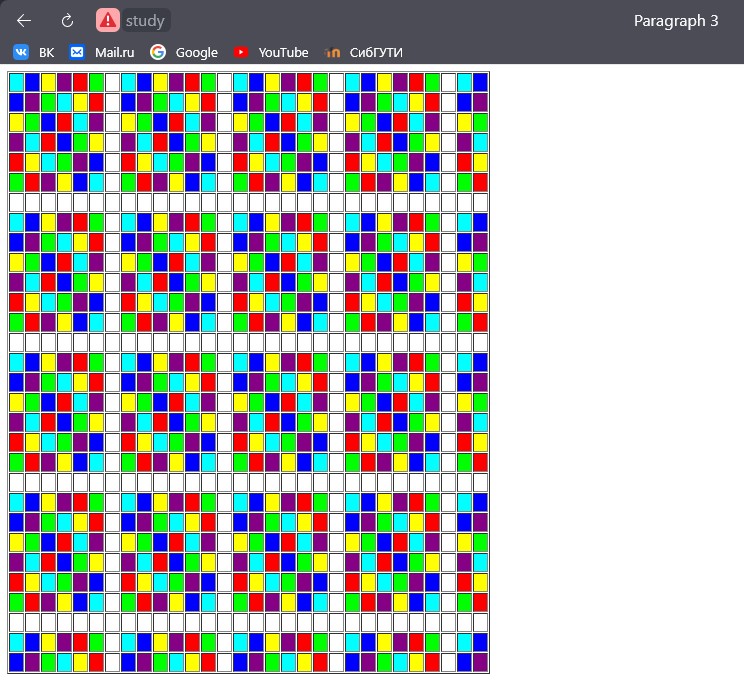
$color = "white";

}

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

} print "</tr>\n"; } print "</table>";

?>



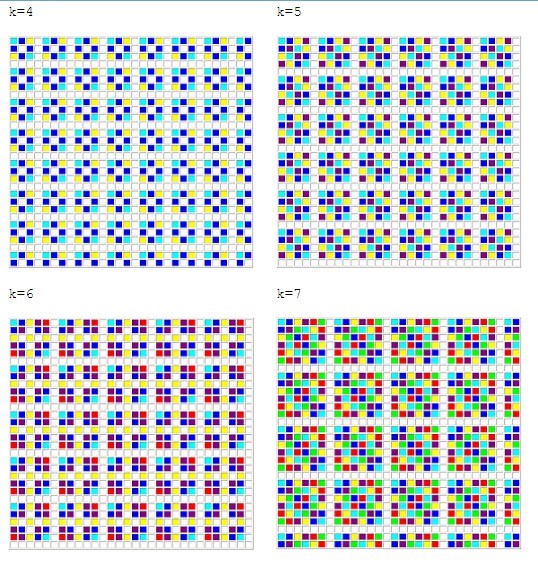
# №4

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).

1. 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).



<?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";

}

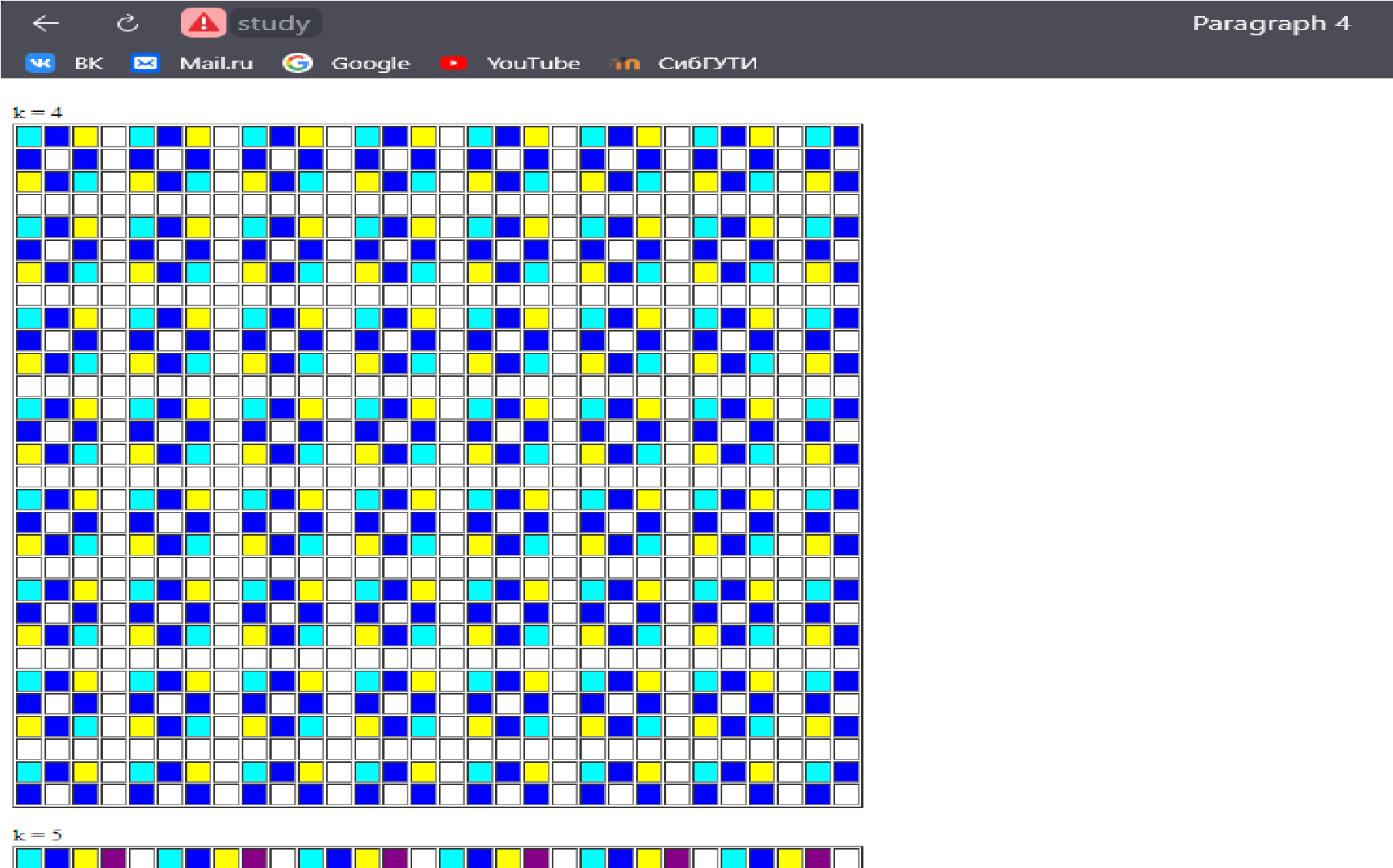
print "\t<td width='14' height='15' bgcolor=$color>"; print ("&nbsp"); print "</td>\n";

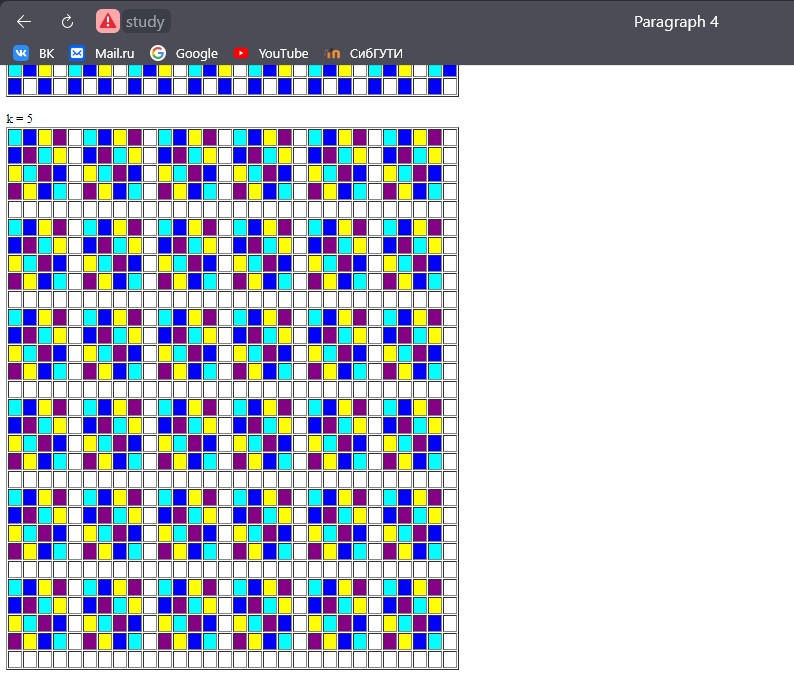
} print "</tr>\n";

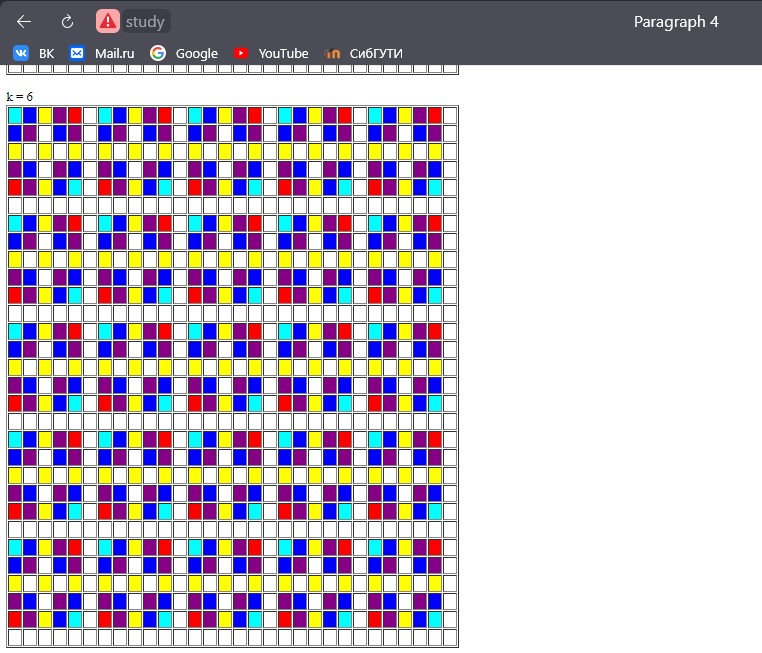
} print "</table>";

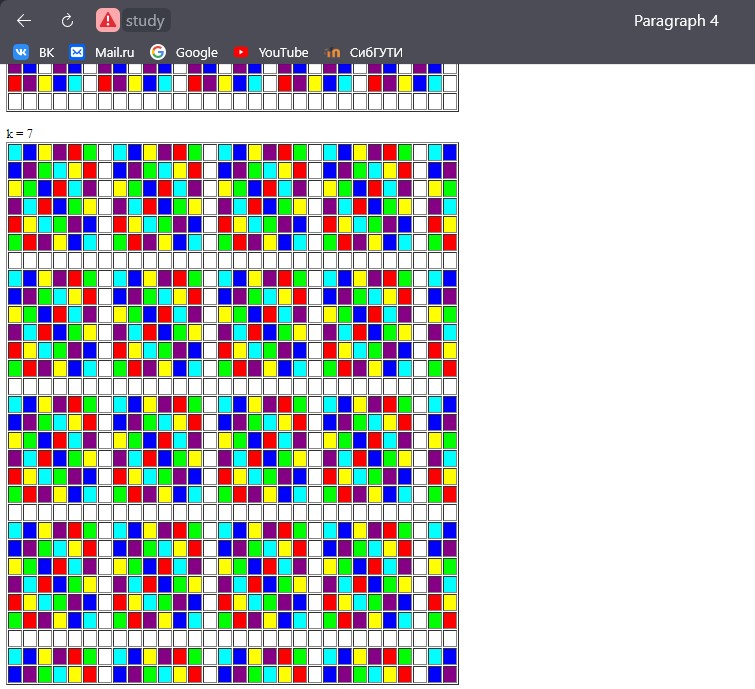
}

?>









# №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 результатна экран.
4. 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>";

}

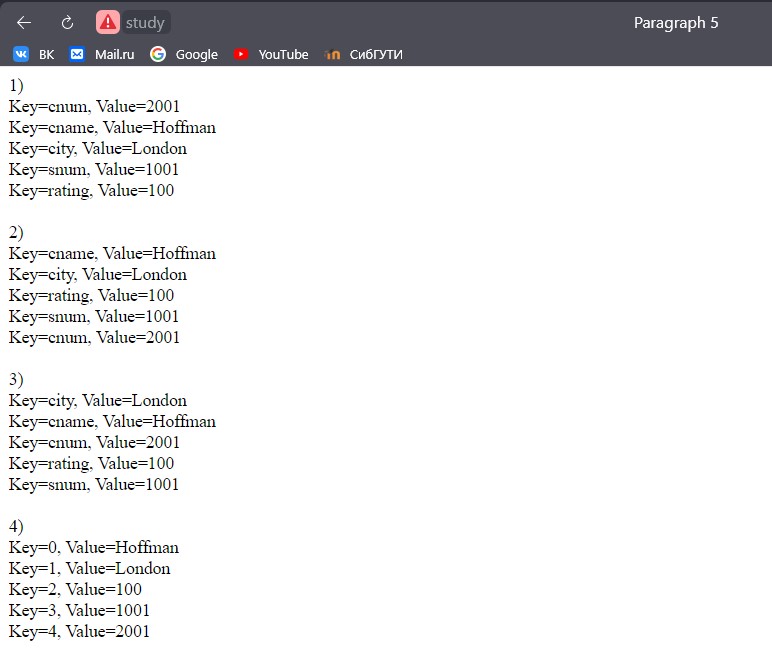
/\*asort($cust); //пример сортировки ассоциативного массива через asort() print("<br> 4) <br>"); foreach($cust as $c => $c\_value) {

echo "Key=" . $c . ", Value=" . $c\_value;

echo "<br>";

}\*/

?>



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.