SQUARES & CUBES

num	square	cube
0	0	0
1	1	1
2	4	8
3	9	27
4	16	64
5	25	125
6	36	216
7	49	343
8	64	512
9	81	729
10	100	1000
11	121	1331
12	144	1728
13	169	2197
14	196	2744
15	225	3375
16	256	4096
17	289	4913
18	324	5832
19	361	6859
20	400	8000

1	2	3	4	5	6	7	8	9	10
1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

1	2	3	4	5	6	7	8	9	10
2	4	6	8	10	12	14	16	18	20
3	6	9	12	15	18	21	24	27	30
4	8	12	16	20	24	28	32	36	40
5	10	15	20	25	30	35	40	45	50
6	12	18	24	30	36	42	48	54	60
7	14	21	28	35	42	49	56	63	70
8	16	24	32	40	48	56	64	72	80
9	18	27	36	45	54	63	72	81	90
10	20	30	40	50	60	70	80	90	100

class.ezpdf.php

```
<?php

include_once('class.pdf.php');

class Cezpdf extends Cpdf {
//============================================================================
// this class will take the basic interaction facilities of the Cpdf class
// and make more useful functions so that the user does not have to
// know all the ins and outs of pdf presentation to produce something pretty.
//
// IMPORTANT NOTE
// there is no warranty, implied or otherwise with this software.
//
// version 003 (versioning is linked to class.pdf.php)
//
// Wayne Munro, R&OS Ltd, 4 December 2001
//============================================================================

var $ez=array('fontSize'=>10); // used for storing most of the page configuration parameters
var $y; // this is the current vertical positon on the page of the writing point, very important

function Cezpdf($paper='a4',$orientation='portrait'){
  // Assuming that people don't want to specify the paper size using the absolute coordinates
  // allow a couple of options:
  // paper can be 'a4' or 'letter'
  // orientation can be 'portrait' or 'landscape'
  // or, to actually set the coordinates, then pass an array in as the first parameter.
  // the defaults are as shown.

  if (!is_array($paper)){
    switch (strtolower($paper)){
      case 'letter':
        $size = array(0,0,612,792);
        break;
      case 'a4':
      default:
        $size = array(0,0,598,842);
        break;
    }
    switch (strtolower($orientation)){
      case 'landscape':
        $a=$size[3];
        $size[3]=$size[2];
        $size[2]=$a;
        break;
    }
  } else {
    // then an array was sent it to set the size
    $size = $paper;
  }
  $this->Cpdf($size);
  $this->ez['pageWidth']=$size[2];
  $this->ez['pageHeight']=$size[3];

  // also set the margins to some reasonable defaults
  $this->ez['topMargin']=30;
  $this->ez['bottomMargin']=30;
  $this->ez['leftMargin']=30;
  $this->ez['rightMargin']=30;

  // set the current writing position to the top of the first page
  $this->y = $this->ez['pageHeight']-$this->ez['topMargin'];
}

function ezNewPage(){
  // make a new page, setting the writing point back to the top
  $this->y = $this->ez['pageHeight']-$this->ez['topMargin'];
  // make the new page with a call to the basic class.
  $this->newPage();
}

function ezSetMargins($top,$bottom,$left,$right){
  // sets the margins to new values
  $this->ez['topMargin']=$top;
  $this->ez['bottomMargin']=$bottom;
  $this->ez['leftMargin']=$left;
  $this->ez['rightMargin']=$right;
  // check to see if this means that the current writing position is outside the
  // writable area
  if ($this->y > $this->ez['pageHeight']-$top){
    // then move y down
    $this->y = $this->ez['pageHeight']-$top;
  }
  if ( $this->y < $bottom){
    // then make a new page
    $this->ezNewPage();
  }
}

function ezSetY($y){
  // used to change the vertical position of the writing point.
  $this->y = $y;
  if ( $this->y < $this->ez['marginBottom']){
    // then make a new page
    $this->ezNewPage();
  }
}

function ezSetDy($dy){
  // used to change the vertical position of the writing point.
  // changes up by a positive increment, so enter a negative number to go
  // down the page
  $this->y += $dy;
  if ( $this->y < $this->ez['marginBottom']){
    // then make a new page
    $this->ezNewPage();
  }
}
```

```
num
109
110  function ezPrvtTableDrawLines($pos,$gap,$x0,$x1,$y0,$y1,$y2,$col){
111    $x0=1000;
112    $x1=0;
113    $this->setStrokeColor($col[0],$col[1],$col[2]);
114  //  $pdf->setStrokeColor(0.8,0.8,0.8);
115    foreach($pos as $x){
116      $this->line($x-$gap/2,$y0,$x-$gap/2,$y2);
117      if ($x>$x1){ $x1=$x; };
118      if ($x<$x0){ $x0=$x; };
119    }
120    $this->line($x0-$gap/2,$y0,$x1-$gap/2,$y0);
121    if ($y0!=$y1){
122      $this->line($x0-$gap/2,$y1,$x1-$gap/2,$y1);
123    }
124    $this->line($x0-$gap/2,$y2,$x1-$gap/2,$y2);
125  }
126
127  function ezPrvtTableColumnHeadings($cols,$pos,$height,$gap,$size,&$y){
128    $y=$y-$height;
129    foreach($cols as $colName=>$colHeading){
130      $this->addText($pos[$colName],$y,$size,$colHeading);
131    }
132    $y -= $gap;
133  }
134
135  function ezTable(&$data,$cols='',$title='',$options=''){
136    // add a table of information to the pdf document
137    // $data is a two dimensional array
138    // $cols (optional) is an associative array, the keys are the names of the columns from $data
139    // to be presented (and in that order), the values are the titles to be given to the columns
140    // $title (optional) is the title to be put on the top of the table
141    //
142    // $options is an associative array which can contain:
143    // 'showLines'=> 0 or 1, default is 1 (1->alternate lines are shaded, 0->no shading)
144    // 'showHeadings' => 0 or 1
145    // 'shaded'=> 0 or 1, default is 1 (1->alternate lines are shaded, 0->no shading)
146    // 'shadeCol' => (r,g,b) array, defining the colour of the shading, default is (0.8,0.8,0.8)
147    // 'fontSize' => 10
148    // 'textCol' => (r,g,b) array, text colour
149    // 'titleFontSize' => 12
150    // 'titleGap' => 5 , the space between the title and the top of the table
```

This is a length of text to test the functioning of the new justification features of the pdf class - hopefully this text will all be placed on the page exactly as required.
The quick brown fox jumped over the lazy dog, who knows why? Anyway, if this is working correctly, then there should be four distinct blocks of text, the first left justified, the second right, the third centered and the fourth with full justification

This is a length of text to test the functioning of the new justification features of the pdf class - hopefully this text will all be placed on the page exactly as required.
The quick brown fox jumped over the lazy dog, who knows why? Anyway, if this is working correctly, then there should be four distinct blocks of text, the first left justified, the second right, the third centered and the fourth with full justification

This is a length of text to test the functioning of the new justification features of the pdf class - hopefully this text will all be placed on the page exactly as required.
The quick brown fox jumped over the lazy dog, who knows why? Anyway, if this is working correctly, then there should be four distinct blocks of text, the first left justified, the second right, the third centered and the fourth with full justification

all be placed on the page exactly as required.

[illegible]

justification

second right, the third centered and the fourth with full