Looks great. 8 / 8

PS 11— Rania

Section 29

In[453]:=

```
(* 27.1 Use Prime and Array to generate a list of the first 100 primes.*)
Array[Prime, 100]
```

Out[453]=

```
{2, 3, 5, 7, 11, 13, 17, 19, 23, 29, 31, 37, 41, 43, 47, 53, 59, 61, 67, 71, 73, 79,
 83, 89, 97, 101, 103, 107, 109, 113, 127, 131, 137, 139, 149, 151, 157, 163,
 167, 173, 179, 181, 191, 193, 197, 199, 211, 223, 227, 229, 233, 239, 241, 251,
 257, 263, 269, 271, 277, 281, 283, 293, 307, 311, 313, 317, 331, 337, 347, 349,
 353, 359, 367, 373, 379, 383, 389, 397, 401, 409, 419, 421, 431, 433, 439,
 443, 449, 457, 461, 463, 467, 479, 487, 491, 499, 503, 509, 521, 523, 541}
```

In[454]:=

```
(*29.2 Use Prime and Array to find successive
 differences between the first 100 primes.*)
Array[Prime[# + 1] - Prime[#] &, 99]
```

Out[454]=

```
{1, 2, 2, 4, 2, 4, 2, 4, 6, 2, 6, 4, 2, 4, 6, 6, 2, 6, 4, 2, 6, 4, 6, 8, 4, 2,
 4, 2, 4, 14, 4, 6, 2, 10, 2, 6, 6, 4, 6, 6, 2, 10, 2, 4, 2, 12, 12, 4, 2, 4,
 6, 2, 10, 6, 6, 6, 2, 6, 4, 2, 10, 14, 4, 2, 4, 14, 6, 10, 2, 4, 6, 8, 6, 6,
 4, 6, 8, 4, 8, 10, 2, 10, 2, 6, 4, 6, 8, 4, 2, 4, 12, 8, 4, 8, 4, 6, 12, 2, 18}
```

In[455]:=

```
(* 29.3 Use Array and Grid to make a 10 by 10 addition table.*)
Array[Times, {10, 10}] // Grid
```

Out[455]=

```
 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
```

In[456]:=

```
(*29.4 Use FoldList,
Times and Range to successively multiply numbers up to 10 (making factorials).*)
FoldList[Times, 1, Range[10]]
(*a little confused tbh about this format*)
```

Well, you managed to make it work.

Out[456]=

```
{1, 1, 2, 6, 24, 120, 720, 5040, 40 320, 362 880, 3 628 800}
```

In[457]:=

```
(*29.5 Use FoldList and Array to compute
 the successive products of the first 10 primes.*)
FoldList[Times, 1 , Array[Prime, 10]]
```

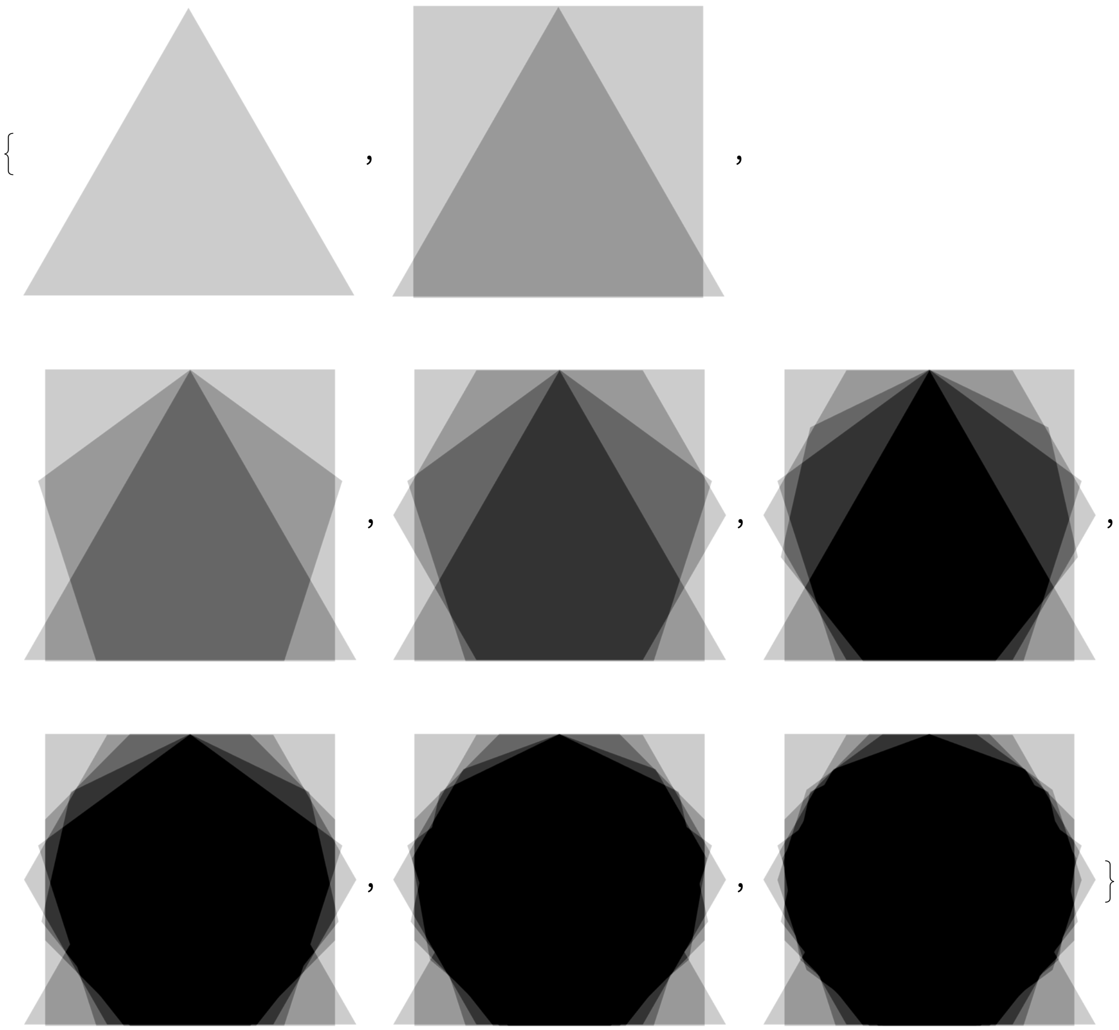
Out[457]=

```
{1, 2, 6, 30, 210, 2310, 30 030, 510 510, 9 699 690, 223 092 870, 6 469 693 230}
```

In[458]:=

```
(*29.6 Use FoldList to successively ImageAdd regular
 polygons with between 3 and 8 sides,and with opacity 0.2*)
FoldList[ImageAdd,
 Table[Graphics[Style[RegularPolygon[n], Opacity[0.2]]], {n, 3, 10}]]
```

Out[458]=



Section 30

In[459]:=

```
(*30.1 Use Thread to make a list of rules with each
 letter of the alphabet going to its position in the alphabet.*)
Thread[Alphabet[] → LetterNumber[Alphabet[]]]
```

Out[459]=

```
{a → 1, b → 2, c → 3, d → 4, e → 5, f → 6, g → 7, h → 8,
 i → 9, j → 10, k → 11, l → 12, m → 13, n → 14, o → 15, p → 16, q → 17,
 r → 18, s → 19, t → 20, u → 21, v → 22, w → 23, x → 24, y → 25, z → 26}
```

In[460]:=

```
(*30.2 Make a 4x6 grid of the first 24 letters of the alphabet.*)
Grid[Partition[Alphabet[], 6]]
```

Out[460]=

```
a b c d e f
g h i j k l
m n o p q r
s t u v w x
```

In[461]:=

```
(*30.3 Make a grid of the digits in 2^1000,
with 50 digits per row,and put frames around everything*)
Grid[Partition[IntegerDigits[2^1000], 50], Frame → All]
```

Out[461]=

```
|1|0|7|1|5|0|8|6|0|7|1|8|6|2|6|7|3|2|0|9|4|8|4|2|5|0|4|9|0|6|0|0|0|1|8|1|0|5|6|1|4|0|4|8|1|1|7
|3|3|6|0|7|4|4|3|7|5|0|3|8|8|3|7|0|3|5|1|0|5|1|1|2|4|9|3|6|1|2|2|4|9|3|1|9|8|3|7|8|8|1|5|6|9|5
|1|2|7|5|9|4|6|7|2|9|1|7|5|5|3|1|4|6|8|2|5|1|8|7|1|4|5|2|8|5|6|9|2|3|1|4|0|4|3|5|9|8|4|5|7|7|5
|9|8|5|7|4|8|0|3|9|3|4|5|6|7|7|7|4|8|2|4|2|3|0|9|8|5|4|2|1|0|7|4|6|0|5|0|6|2|3|7|1|1|4|1|8|7|7
|1|8|2|1|5|3|0|4|6|4|7|4|9|8|3|5|8|1|9|4|1|2|6|7|3|9|8|7|6|7|5|5|9|1|6|5|5|4|3|9|4|6|0|7|7|0|6
|4|5|7|1|1|9|6|4|7|7|6|8|6|5|4|2|1|6|7|6|6|0|4|2|9|8|3|1|6|5|2|6|2|4|3|8|6|8|3|7|2|0|5|6|6|8|0
```

In[462]:=

```
(*30.4 Make a grid of the first 400 characters in the Wikipedia article for
 “computers”,with 20 characters per row,and frames around everything.*)
Grid[Partition[Take[Characters[WikipediaData["computer"]], 400], 20], Frame → All]
```

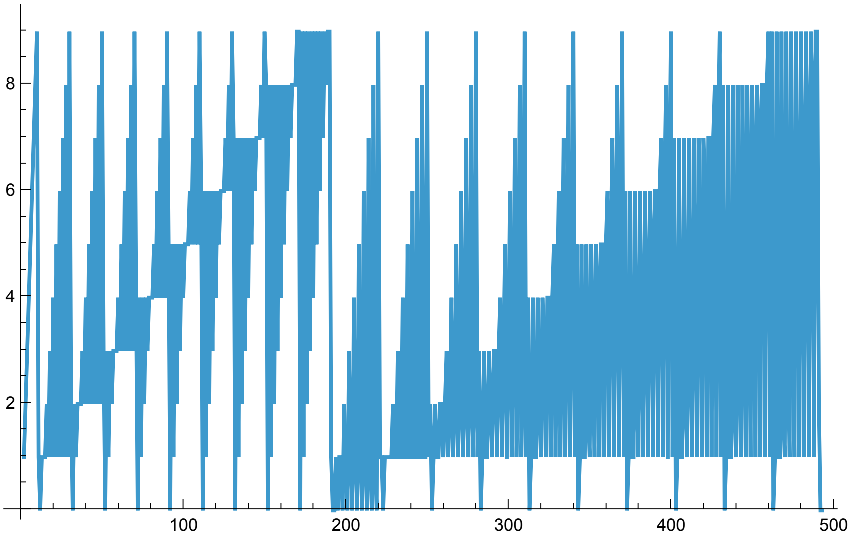
Out[462]=

A		c	o	m	p	u	t	e	r		i	s		a		m	a	c	h
i	n	e		t	h	a	t		c	a	n		b	e		p	r	o	g
r	a	m	m	e	d		t	o		a	u	t	o	m	a	t	i	c	a
l	l	y		c	a	r	r	y		o	u	t		s	e	q	u	e	n
c	e	s		o	f		a	r	i	t	h	m	e	t	i	c		o	r
	l	o	g	i	c	a	l		o	p	e	r	a	t	i	o	n	s	
(	c	o	m	p	u	t	a	t	i	o	n	)	.		M	o	d	e	r
n		d	i	g	i	t	a	l		e	l	e	c	t	r	o	n	i	c
	c	o	m	p	u	t	e	r	s		c	a	n		p	e	r	f	o
r	m		g	e	n	e	r	i	c		s	e	t	s		o	f		o
p	e	r	a	t	i	o	n	s		k	n	o	w	n		a	s		p
r	o	g	r	a	m	s	.		T	h	e	s	e		p	r	o	g	r
a	m	s		e	n	a	b	l	e		c	o	m	p	u	t	e	r	s
	t	o		p	e	r	f	o	r	m		a		w	i	d	e		r
a	n	g	e		o	f		t	a	s	k	s	.		T	h	e		t
e	r	m		c	o	m	p	u	t	e	r		s	y	s	t	e	m	
m	a	y		r	e	f	e	r		t	o		a		n	o	m	i	n
a	l	l	y		c	o	m	p	l	e	t	e		c	o	m	p	u	t
e	r		t	h	a	t		i	n	c	l	u	d	e	s		t	h	e
	h	a	r	d	w	a	r	e	,		o	p	e	r	a	t	i	n	g

In[463]:=

```
(*30.5 Make a line plot of the flattened list of the digits
 from the numbers from 0 to 200 (Champernowne sequence).*)
ListLinePlot[Flatten[IntegerDigits /@ Range[200]]]
```

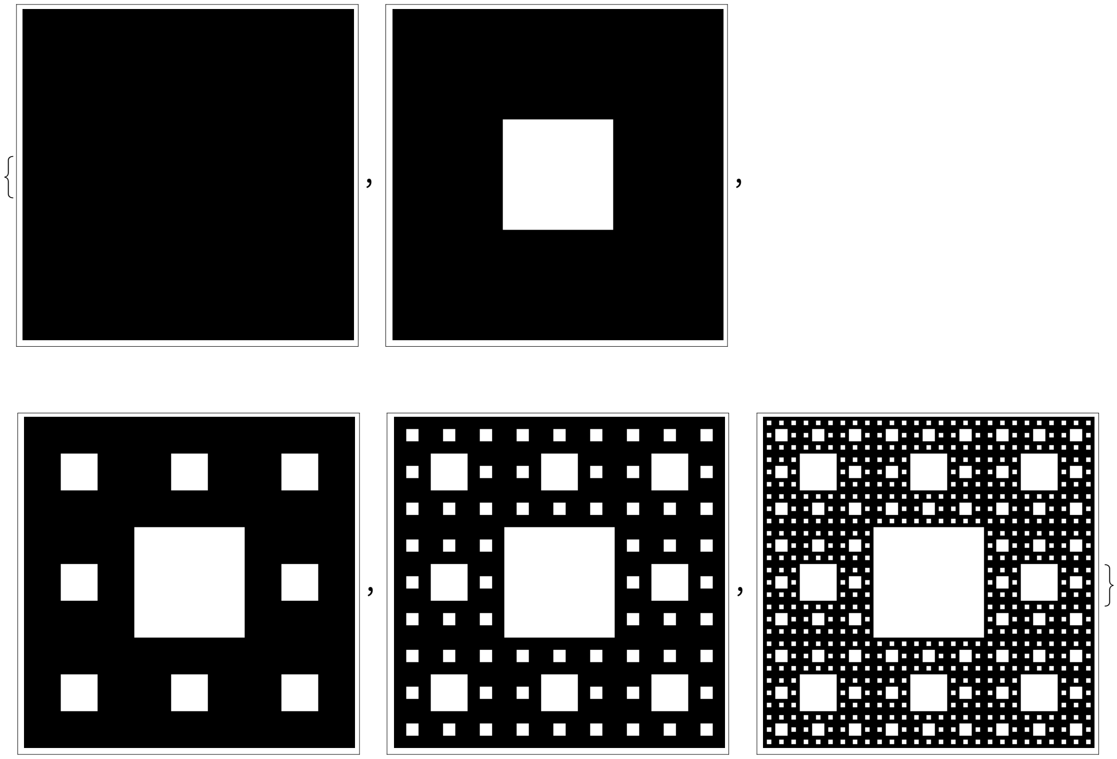
Out[463]=



In[464]:=

```
(*30.6 Make 4 steps in the “Menger sponge”
 analog of the fractal Sierpinski pattern from the text,
but with a “kernel” of the form {{#,#,#},{#,0,#},{#,#,#}}.*)
ArrayPlot /@ NestList[ArrayFlatten[{{#, #, #}, {#, 0, #}, {#, #, #}}] &, {{1}}, 4]
```

Out[464]=



In[465]:=

```
(*30.7 Find Pythagorean triples involving only integers by
 selecting {x,y,Sqrt[x^2+y^2]} with x and y up to 20.*)Select[
 Flatten[Table[{x, y, Sqrt[x^2 + y^2]}, {x, 20}, {y, 20}], 1], IntegerQ[Last[#]] &]
```

Out[465]=

```
{{3, 4, 5}, {4, 3, 5}, {5, 12, 13}, {6, 8, 10},
 {8, 6, 10}, {8, 15, 17}, {9, 12, 15}, {12, 5, 13}, {12, 9, 15},
 {12, 16, 20}, {15, 8, 17}, {15, 20, 25}, {16, 12, 20}, {20, 15, 25}}
```

In[466]:=

```
(*30.8 Find the lengths of the longest sequences
 of identical digits in 2^n for n up to 100. »*)
Max /@ (Length /@ Split[IntegerDigits[2^#]] & /@ Range[100])
```

Out[466]=

```
{1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 2, 1, 2, 2, 1, 1, 1, 2, 3,
 2, 2, 2, 1, 1, 1, 1, 1, 2, 1, 1, 1, 1, 2, 3, 3, 4, 3, 3, 3, 3, 2, 2, 1, 2,
 3, 2, 2, 2, 1, 1, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1, 1, 1, 1, 2, 2, 2, 3, 3,
 3, 3, 3, 2, 2, 1, 2, 2, 3, 2, 2, 2, 1, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2}
```

```
In[467]:=
(*30.9 Take the names of integers up to 100 and gather
 them into sublists according to their first letters.*)
GatherBy[IntegerName[#] & /@ Range[100], StringTake[#, 1] &]
Out[467]=
{{one, one hundred}, {two, three, ten, twelve, thirteen, twenty, twenty-one,
  twenty-two, twenty-three, twenty-four, twenty-five, twenty-six, twenty-seven,
  twenty-eight, twenty-nine, thirty, thirty-one, thirty-two, thirty-three,
  thirty-four, thirty-five, thirty-six, thirty-seven, thirty-eight, thirty-nine},
 {four, five, fourteen, fifteen, forty, forty-one, forty-two, forty-three,
  forty-four, forty-five, forty-six, forty-seven, forty-eight,
  forty-nine, fifty, fifty-one, fifty-two, fifty-three, fifty-four,
  fifty-five, fifty-six, fifty-seven, fifty-eight, fifty-nine},
 {six, seven, sixteen, seventeen, sixty, sixty-one, sixty-two, sixty-three,
  sixty-four, sixty-five, sixty-six, sixty-seven, sixty-eight, sixty-nine,
  seventy, seventy-one, seventy-two, seventy-three, seventy-four,
  seventy-five, seventy-six, seventy-seven, seventy-eight, seventy-nine},
 {eight, eleven, eighteen, eighty, eighty-one, eighty-two, eighty-three,
  eighty-four, eighty-five, eighty-six, eighty-seven, eighty-eight, eighty-nine},
 {nine, nineteen, ninety, ninety-one, ninety-two, ninety-three, ninety-four,
  ninety-five, ninety-six, ninety-seven, ninety-eight, ninety-nine}}
```

```
In[468]:=
(*30.10 Sort the first 50 words in WordList[] by their last letters.*)

SortBy[Take[WordList[], 50], StringTake[StringReverse[#], 1] &]
Out[468]=
{a, abandoned, abashed, abbreviated, abed, abalone, abase, abate, abbe, abbreviate,
 abdicate, abeyance, abhorrence, abidance, abide, abducting, abiding, aah,
 abash, aardvark, aback, abdominal, abeam, abandon, abbreviation, abdication,
 abdomen, abduction, aberration, abjection, abattoir, abductor, abettor,
 abhor, abacus, abbess, abaft, abandonment, abasement, abashment, abatement,
 abbot, abduct, aberrant, abet, abhorrent, abject, abbey, ability, abjectly}
```

```
In[469]:=
(*30.11 Make a list of the first 20 squares,sorted by their first digits.*)
SortBy[#^2 & /@ Range[20], First[IntegerDigits[#]] &]
Out[469]=
{1, 16, 100, 121, 144, 169, 196, 25, 225, 256, 289, 36, 324, 361, 4, 49, 400, 64, 81, 9}
```

```
In[470]:=
(*30. 11 Sort integers up to 20 by the length of their names in English*)
SortBy[# & /@ Range[20], StringLength[IntegerName[#]] &]
Out[470]=
{1, 2, 6, 10, 4, 5, 9, 3, 7, 8, 11, 12, 20, 15, 16, 13, 14, 18, 19, 17}
```

In[471]:=

```
(*30.13 Get a random sample of 20 words from WordList[],
and gather them into sublists by length.*)
GatherBy[RandomSample[WordList[], 20], StringLength[#] &]
```

Out[471]=

```
{{gracious, politics, humpback, outclass},
 {benignity, pampering, contorted, patrolman, worrisome, perfecter},
 {aforethought}, {anomaly, caldera}, {open, java},
 {historiography}, {screw, bring}, {gem}, {monitoring}}
```

In[472]:=

```
(*30.14 Find letters that appear in Ukrainian but not Russian.*)
Complement[Alphabet["Ukrainian"], Alphabet["Russian"]]
```

Out[472]=

```
{є, і, ї, ґ}
```

In[473]:=

```
(*30.15 Use Intersection to find numbers that
 appear both among the first 100 squares and cubes.*)
Intersection[Power[Range[100], 2], Power[Range[100], 3]]
```

Out[473]=

```
{1, 64, 729, 4096}
```

In[474]:=

```
(*30.16 Find the list of countries that are in both NATO and the G8.*)
```

In[475]:=

```
Intersection[EntityList[ North Atlantic Treaty Organization COUNTRIES ],
 EntityList[ Group of 8 COUNTRIES ]]
```

Out[475]=

```
{ Canada , France , Germany , Italy , United Kingdom , United States }
```

In[476]:=

```
(*30.1 7Make a grid in which all possible permutations
 of the numbers 1 through 4 appear as successive columns.*)
Grid[Transpose[Permutations[Range[4]]]]
```

Out[476]=

```
1 1 1 1 1 1 2 2 2 2 2 2 3 3 3 3 3 3 4 4 4 4 4 4
2 2 3 3 4 4 1 1 3 3 4 4 1 1 2 2 4 4 1 1 2 2 3 3
3 4 2 4 2 3 3 4 1 4 1 3 2 4 1 4 1 2 2 3 1 3 1 2
4 3 4 2 3 2 4 3 4 1 3 1 4 2 4 1 2 1 3 2 3 1 2 1
```

In[477]:=

```
(*30.18Make a list of all the different strings that
 can be obtained by permuting the characters in “hello”.*)
StringJoin /@ Permutations[Characters["hello"]]
```

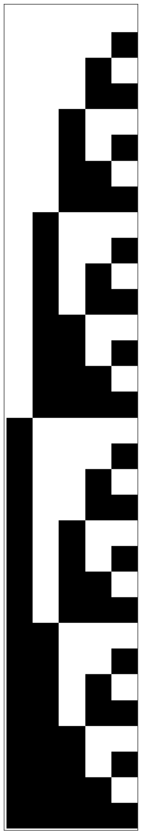
Out[477]=

```
{hello, helol, heoll, hlelo, hleol, hlleo, hlloe, hloel, hlole, hoell, holel, holle,
 ehllo, ehlol, eholl, elhlo, elhol, ellho, elloh, elohl, elolh, eohll, eolhl, eollh,
 lhelo, lheol, lhleo, lhloe, lhoel, lhole, lehlo, lehol, lelho, leloh, leohl, leolh,
 llheo, llhoe, lleho, lleoh, llohe, lloeh, lohel, lohle, loehl, loelh, lolhe, loleh,
 ohell, ohlel, ohlle, oehll, oelhl, oellh, olhel, olhle, olehl, olelh, ollhe, olleh}
```

In[478]:=

```
(*30.19Make an array plot of the sequence of possible 5-tuples of 0 and 1.*)
ArrayPlot[Tuples[{0, 1}, 5]]
```

Out[478]=



In[479]:=

```
(*30.20Generate a list of 10 random sequences of 5 letters.*)
Table[StringJoin[RandomChoice[Alphabet[], 5]] , 10]
```

Out[479]=

```
{eiepd, zlkhc, zjrbt, lprfg, vwxsp, pfpdc, xgsxv, gljbz, yqqdo, uldii}
```

```
In[480]:=
        (*30.21 Find a simpler form for Flatten[Table[{i,j,k},{i,2},{j,2},{k,2}],2].*)
        Flatten[Table[{i, j, k}, {i, 2}, {j, 2}, {k, 2}], 2]
        Tuples[{1, 2}, 3]
Out[480]=
        {{1, 1, 1}, {1, 1, 2}, {1, 2, 1}, {1, 2, 2}, {2, 1, 1}, {2, 1, 2}, {2, 2, 1}, {2, 2, 2}}
Out[481]=
        {{1, 1, 1}, {1, 1, 2}, {1, 2, 1}, {1, 2, 2}, {2, 1, 1}, {2, 1, 2}, {2, 2, 1}, {2, 2, 2}}
```