Problems from Section 4.1

Problem 20

■ (a) How many integers in {1000,...,9999} are divisable by 9

We have 999 = 111*9. Thus the smallest integer in this set divisable by 9 is 112*9. The largest integer divisable by 9 in this set is 9999=1111*9. Thus there are

```
1111 - 111
```

```
1000
```

integers in the given set whichare divisable by 9

We can verify this in *Mathematica* by the following commands

```
IntegerQ[#/9] &[7866]
```

```
True
```

```
Length[Select[Range[1000, 9999], IntegerQ[#/9] &]]
```

```
1000
```

■ (b) How many integers in {1000,...,9999} are even

1000 = 500*2 is even and 4999*2 = 9998 is even. Thus, there are

```
4999 - 500 + 1
```

```
4500
```

even integers in this set.

Verification in *Mathematica*

```
Length[Select[Range[1000, 9999], EvenQ[#] &]]
```

```
4500
```

■ (c) How many integers in {1000,...,9999} have distinct digits

We calculate this by using the product rule. There are 9 options, that is {1, 2, 3, 4, 5, 6, 7, 8, 9}, to choose the first digit; say this is d_1; there are 9 options, that is {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\{d_1}, to choose the second digit; say this is d_2; there are 8 options, that is {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\{d_1, d_2}, to choose the third digit; say this is d_3; and finally, there are 7 options, that is {0, 1, 2, 3, 4, 5, 6, 7, 8, 9}\{d_1, d_2, d_3}, to choose the fourth digit.

```
9 9 8 7
```

```
4536
```

Verification in *Mathematica*

The command

```
Length[Union[IntegerDigits[#]]] &[3456]
```

```
4
```

tells us how many DISTINCT digits there are in an integer. The command

```
Length[Union[IntegerDigits[#]]] == 4 &[3456]
```

```
True
```

tells if the number of DISTINCT digits is equal to 4

Finally this is *Mathematica*'s answer to (c)

```
Length[Select[Range[1000, 9999], Length[Union[IntegerDigits[#]]] == 4 &]]
```

```
4536
```

■ (d) How many integers in {1000,...,9999} are not divisable by 3

999=333*3 and 9999=3333*3, so there are 3000 integers divisable by 3. Since the total is 9000 integers, there are 6000 integers not divisable by 3

```
Length[Select[Range[1000, 9999], Not[IntegerQ[#/3]] &]]
```

```
6000
```

■ (e), (f), (g), (h)

The relevant counts are:

the number of integers divisable by 5

```
div5 = Floor[9999/5] - Floor[999/5]
```

$$\texttt{div5} = \texttt{Floor}\left[\frac{9999}{5}\right] - \texttt{Floor}\left[\frac{999}{5}\right]$$

```
1800
```

the number of integers divisable by 7

$$\texttt{div7} = \texttt{Floor}\left[\frac{9999}{7}\right] - \texttt{Floor}\left[\frac{999}{7}\right]$$

```
1286
```

the number of integers divisable by both 5 and 7

$$\texttt{div35} = \texttt{Floor}\left[\frac{9999}{35}\right] - \texttt{Floor}\left[\frac{999}{35}\right]$$

```
General::spell1 :
 Possible spelling error: new symbol name "div35" is similar to existing symbol "div5". More…
```

```
257
```

So, the answer to (e), the number of integers divisable by 5 or 7 is

```
div5 + div7 - div35
```

```
2829
```

Mathematica verification

$$\texttt{Length}\left[\texttt{Select}\left[\texttt{Range}[1000, 9999], \texttt{Or}\left[\texttt{IntegerQ}\left[\frac{\#}{7}\right], \texttt{IntegerQ}\left[\frac{\#}{5}\right]\right] \&\right]\right]$$

```
2829
```

The answer to (f),the number of integers not divisable by either 5 or 7 is

```
9000 - (div5 + div7 - div35)
```

```
6171
```

Mathematica verification

$$\texttt{Length}\left[\texttt{Select}\left[\texttt{Range}[1000, 9999], \texttt{Not}\left[\texttt{Or}\left[\texttt{IntegerQ}\left[\frac{\#}{7}\right], \texttt{IntegerQ}\left[\frac{\#}{5}\right]\right]\right] \&\right]\right]$$

```
6171
```

The answer to (g), the number of integers divisable by 5 but not divisable by 7 is

```
div5 - div35
```

```
1543
```

Mathematica verification

```
Length[Select[Range[1000, 9999], And[Not[IntegerQ[#/7]], IntegerQ[#/5]] &]]
```

```
1543
```

The answer to (h), the number of integers divisable by 5 and divisable by 7 is

```
div35
```

```
257
```

Mathematica verification

```
Length[Select[Range[1000, 9999], And[IntegerQ[#/7], IntegerQ[#/5]] &]]
```

```
257
```

To verify problems involving strings we need

```
<< DiscreteMath`Combinatorica`
```

```
Strings[{0, 1}, 4]
```

```
{{0, 0, 0, 0}, {0, 0, 0, 1}, {0, 0, 1, 0}, {0, 0, 1, 1}, {0, 1, 0, 0},
 {0, 1, 0, 1}, {0, 1, 1, 0}, {0, 1, 1, 1}, {1, 0, 0, 0}, {1, 0, 0, 1},
 {1, 0, 1, 0}, {1, 0, 1, 1}, {1, 1, 0, 0}, {1, 1, 0, 1}, {1, 1, 1, 0}, {1, 1, 1, 1}}
```

Problem 40

How many bit strings of length 7 start with 00 or end with 111?

There are 32 strings which start with 00 and there are 16 strings which end with 111. There are 4 strings that start with 00 and end with 111. Thus the answer is by inclusion-exclusion rule

```
32 + 16 - 4
```

```
44
```

Mathematica verification

The following command will tell me which is the first bit in a bitstring

```
#[[1]] &[{1, 0, 0, 1, 0, 1, 1}]
```

```
1
```

The following command will tell me what problem is asking for

```
Or[And[#[[1]] == 0, #[[2]] == 0], And[#[[7]] == 1, #[[6]] == 1, #[[5]] == 1]] &[{1, 0, 1, 0, 0, 1, 1}]
```

```
False
```

```
Or[And[#[[1]] == 0, #[[2]] == 0], And[#[[7]] == 1, #[[6]] == 1, #[[5]] == 1]] &[{0, 0, 1, 0, 0, 1, 1}]
```

```
True
```

```
Length[Select[Strings[{0, 1}, 7],
  Or[And[#[[1]] == 0, #[[2]] == 0], And[#[[7]] == 1, #[[6]] == 1, #[[5]] == 1]] &]]
```

```
44
```

Problem 42

How many bit strings of length 10 contain at least 5 consecutive 0s or at least 5 consecutive 1s?

We will first count the bitstrings with at least 5 consecutive 0s.

The "at least 5 consecutive 0s" can start at the following positions 1, 2, 3, 4, 5, 6

There are $2^5 = 32$ bit srings which start with 00000***** (type 1)

There are $2^4 = 16$ bit srings which start with 100000**** (type 2)

There are $2^4 = 16$ bit srings which start with *100000*** (type 3)

There are $2^4 = 16$ bit srings which start with **100000** (type 4)

There are $2^4 = 16$ bit srings which start with ***100000* (type 5)

There are $2^4 = 16$ bit srings which start with ****100000 (type 6)

There are no bitstrings that belong to two different types. (You conclude this by looking at types j and k, j < k. Type k has 1 at position k-1 while type j has 0 at the position k-1.

Thus there are

```
32 + 5 * 16
```

```
112
```

bit strings with at least 5 consecutive 0s

Also, there are 112 bit strings with at least 5 consecutive 1s.

There are 2 bit strings which are in both sets: 0000011111 and 1111100000.

By the inclusion-exclusion principle the answer is

```
2 * 112 - 2
```

```
222
```

To verify this in *Mathematica* is a little bit more complicated.

The following command will collect the identical consecutive bits in separate lists

```
Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]
```

```
{{1}, {0}, {1}, {0, 0}, {1}, {0, 0, 0, 0}}
```

The following command will count how many consecutive bits there are

```
Length[#] & /@ Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]
```

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

The following command will find the maximum number of consecutive bits

```
Max[Length[#] & /@ Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]]
```

```
4
```

And finally we will ask if that max number is ≥5.

```
Max[Length[#] & /@ Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]] ≥ 5
```

```
False
```

Make this into a function of a bit string

```
Max[Length[#] & /@ Split[#]] ≥ 5 &[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]
```

```
False
```

```
Length[Select[Strings[{0, 1}, 10], Max[Length[#] & /@ Split[#]] ≥ 5 &]]
```

```
222
```

Bride and groom

There are 6 people together with the bride and the groom that are about to take a picture. Answer the following questions:

■ Questions

How many different ways that this party can be arranged if they are all to stand next to each other?

The answer is 6! = 720, by the product rule. There are 6 ways to choose the most left person, 5 choices for the next person and so on.

```
allper6 = Permutations[{a, b, c, d, e, f}];
```

```
Length[allper6]
```

```
720
```

How many ways to arrange the party if the bride and the groom are to stand next to each other.

Consider the bride and the groom as one person. There are 5! = 120 ways to arrange. However, in each such arrangment there are two ways to arrange the bride and the groom.

The code below confims that. In this code a is the bride and b is the groom.

```
MemberQ[Partition[{a, b, c, d, e, f}, 2, 1], {a, b}]
```

```
True
```

```
MemberQ[Partition[#, 2, 1], {a, b}] &[{a, b, c, d, e, f}]
```

```
True
```

```
Length[Select[allper6,
  Or[MemberQ[Partition[#, 2, 1], {a, b}], MemberQ[Partition[#, 2, 1], {b, a}]] &]]
```

```
240
```

How many ways to arrange the party if thee is exactly one person between the bride and the groom?

```
4 Length[Select[allper6,
   Or[MemberQ[Partition[#, 3, 1], {a, c, b}], MemberQ[Partition[#, 3, 1], {b, c, a}]] &]]
```

```
192
```

```
Length[Select[allper6,
  Or[MemberQ[Partition[#, 3, 1], {a, c, b}], MemberQ[Partition[#, 3, 1], {b, c, a}],
    MemberQ[Partition[#, 3, 1], {a, d, b}], MemberQ[Partition[#, 3, 1], {b, d, a}],
    MemberQ[Partition[#, 3, 1], {a, e, b}], MemberQ[Partition[#, 3, 1], {b, e, a}],
    MemberQ[Partition[#, 3, 1], {a, f, b}], MemberQ[Partition[#, 3, 1], {b, f, a}]] &]]
```

```
192
```

How many ways to arrange the party in such a way that the bride is to the left of the groom?

```
Select[{b, a, c, d, e, f}, Or[# == a, # == b] &]
```

```
{b, a}
```

```
Function[y, Or[y == a, y == b]][a]
```

```
True
```

```
Function[y, Or[y == a, y == b]][b]
```

```
True
```

```
Select[#, Function[y, Or[y == a, y == b]]] &[{b, a, c, d, e, f}]
```

```
{b, a}
```

```
Length[Select[allper6, (Select[#, Function[y, Or[y == a, y == b]]] == {a, b}) &]]
```

```
360
```