Problems from Section 4.1

■ **To verify problems involving strings we need**

```
In[2]:= << DiscreteMath`Combinatorica`

In[3]:= Strings[{0, 1}, 4]

Out[3]= {{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}}
```

Exercise 11

```
In[4]:= 2^8

Out[4]= 256
```

Exercise 12

```
In[5]:= Sum[2^k, {k, 0, 6}]

Out[5]= 127
```

Exercise 15

```
In[6]:= 26^4 + 26^3 + 26^2 + 26^1 + 1

Out[6]= 475255
```

Exercise 16

```
In[7]:= 26^4 - 25^4

Out[7]= 66351
```

Exercise 17

Inclusion-exclusion

```
In[8]:= 128^5 - 127^5
Out[8]= 1321368961
```

Exercise 18

■ (a)

In[9]:= Floor$\left[\frac{1000}{7}\right]$

Out[9]= 142

■ (b)

In[10]:= Floor$\left[\frac{1000}{11}\right]$

Out[10]= 90

In[11]:= Floor$\left[\frac{1000}{7\ 11}\right]$

Out[11]= 12

```
In[12]:= 142 - 12
Out[12]= 130
```

In[13]:= Length$\left[\text{Select}\left[\text{Range}[999], \text{And}\left[\text{IntegerQ}\left[\frac{\#}{7}\right], \text{Not}\left[\text{IntegerQ}\left[\frac{\#}{11}\right]\right]\right] \&\right]\right]$

Out[13]= 130

■ (c)

In[14]:= Floor$\left[\frac{1000}{11 * 7}\right]$

Out[14]= 12

■ (d)

In[15]:= $\text{Floor}\left[\frac{1000}{11}\right] + \text{Floor}\left[\frac{1000}{7}\right] - \text{Floor}\left[\frac{1000}{11*7}\right]$

Out[15]= 220

■ (e)

In[16]:= $\text{Floor}\left[\frac{1000}{11}\right] + \text{Floor}\left[\frac{1000}{7}\right] - 2\,\text{Floor}\left[\frac{1000}{11*7}\right]$

Out[16]= 208

■ (f)

In[17]:= $999 - \left(\text{Floor}\left[\frac{1000}{11}\right] + \text{Floor}\left[\frac{1000}{7}\right] - \text{Floor}\left[\frac{1000}{11*7}\right]\right)$

Out[17]= 779

■ (g)

```
In[18]:= 90 - 9 (* two digit numbers *) + 9 * 9 * 8 (* three digit numbers *)

Out[18]= 729

In[19]:= Length[Select[Table[k, {k, 100, 999}], And[Length[Union[IntegerDigits[#]]] > 2] &]] +
          Length[Select[Table[k, {k, 10, 99}], And[Length[Union[IntegerDigits[#]]] > 1] &]]

Out[19]= 729
```

■ (h)

```
In[20]:= Length[Select[Table[k, {k, 100, 999}],
            And[Length[Union[IntegerDigits[#]]] > 2, EvenQ[#]] &]] + Length[
           Select[Table[k, {k, 10, 99}], And[Length[Union[IntegerDigits[#]]] > 1, EvenQ[#]] &]]

Out[20]= 369

In[21]:= 8 * 8 * 4 (* three digits ending with 2,4,6,8 *) +
          9 * 8 (* three digits ending with 0 *) + 8 * 4
          (* two digits ending with 2,4,6,8 *) + 9 (* two digits ending with 0 *)

Out[21]= 369

In[22]:= Length[Select[Table[k, {k, 100, 999}],
            And[Length[Union[IntegerDigits[#]]] > 2, OddQ[#]] &]] + Length[
           Select[Table[k, {k, 10, 99}], And[Length[Union[IntegerDigits[#]]] > 1, OddQ[#]] &]]

Out[22]= 360
```

```
In[23]:= 8*8*5 (* three digits ending with 1,3,5,7,9 *) +
          8*5 (* two digits ending with 1,3,5,7,9 *)

Out[23]= 360
```

Exercise 19

(a)

```
In[24]:= 105 / 7

Out[24]= 15

In[25]:= 994 / 7

Out[25]= 142

In[26]:= 142 - 15 + 1

Out[26]= 128
```

(b)

```
In[27]:= 450

Out[27]= 450
```

(c)

```
In[28]:= 9

Out[28]= 9
```

(d)

```
In[29]:= 100 / 4

Out[29]= 25

In[30]:= 996 / 4

Out[30]= 249

In[31]:= 249 - 25 + 1

Out[31]= 225

In[32]:= 900 - 225

Out[32]= 675
```

■ (e)

```
In[33]:= (999 - 102) / 3 + 1
Out[33]= 300
In[34]:= 996 / 12 - 108 / 12 + 1
Out[34]= 75
In[35]:= 300 + 225 - 75
Out[35]= 450
```

■ (f)

```
In[36]:= 450
Out[36]= 450
```

■ (g)

```
In[37]:= 300 - 75
Out[37]= 225
```

■ (h)

```
In[38]:= 75
Out[38]= 75
```

Exercise 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

```
In[39]:= 1111 - 111
Out[39]= 1000
```

integers in the given set whichare divisable by 9

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

```
In[40]:= IntegerQ[#/9] &[7866]

Out[40]= True

In[41]:= Length[Select[Range[1000, 9999], IntegerQ[#/9] &]]

Out[41]= 1000
```

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

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

```
In[42]:= 4999 - 500 + 1

Out[42]= 4500
```

even integers in this set.

Verification in *Mathematica*

```
In[43]:= Length[Select[Range[1000, 9999], EvenQ[#] &]]

Out[43]= 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 $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \backslash \{d_1\}$, to choose the second digit; say this is d_2; there are 8 options, that is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \backslash \{d_1, d_2\}$, to choose the third digit; say this is d_3; and finally, there are 7 options, that is $\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\} \backslash \{d_1, d_2, d_3\}$, to choose the fourth digit.

```
In[44]:= 9 9 8 7

Out[44]= 4536
```

Verification in *Mathematica*

The command

```
In[45]:= Length[Union[IntegerDigits[#]]] &[3456]

Out[45]= 4
```

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

```
In[46]:= Length[Union[IntegerDigits[#]]] == 4 &[3456]

Out[46]= True
```

tells if the number of DISTINCT digits is equal to 4

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

```
In[47]:= Length[Select[Range[1000, 9999], Length[Union[IntegerDigits[#]]] == 4 &]]

Out[47]= 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

```
In[48]:= Length[Select[Range[1000, 9999], Not[IntegerQ[#/3]] &]]

Out[48]= 6000
```

(e), (f), (g), (h)

The relevant counts are:

the number of integers divisable by 5

```
In[49]:= div5 = Floor[9999/5] - Floor[999/5]

Out[49]= 1800
```

the number of integers divisable by 7

```
In[50]:= div7 = Floor[9999/7] - Floor[999/7]

Out[50]= 1286
```

the number of integers divisable by both 5 and 7

```
In[51]:= div35 = Floor[9999/35] - Floor[999/35]

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

Out[51]= 257
```

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

```
In[52]:= div5 + div7 - div35

Out[52]= 2829
```

Mathematica verification

```
In[53]:= Length[Select[Range[1000, 9999], Or[IntegerQ[#/7], IntegerQ[#/5]] &]]

Out[53]= 2829
```

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

```
In[54]:= 9000 - (div5 + div7 - div35)

Out[54]= 6171
```

Mathematica verification

```
In[55]:= Length[Select[Range[1000, 9999], Not[Or[IntegerQ[#/7], IntegerQ[#/5]]] &]]

Out[55]= 6171
```

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

```
In[56]:= div5 - div35

Out[56]= 1543
```

Mathematica verification

```
In[57]:= Length[Select[Range[1000, 9999], And[Not[IntegerQ[#/7]], IntegerQ[#/5]] &]]

Out[57]= 1543
```

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

```
In[58]:= div35

Out[58]= 257
```

Mathematica verification

```
In[59]:= Length[Select[Range[1000, 9999], And[IntegerQ[#/7], IntegerQ[#/5]] &]]

Out[59]= 257
```

Exercise 21

(a)

```
In[60]:= 10^3 - 10
Out[60]= 990
```

(b)

```
In[61]:= 5*10*10
Out[61]= 500
```

(c)

```
In[62]:= 9+9+9
Out[62]= 27
```

Exercise 23

```
In[63]:= 3^50
Out[63]= 717897987691852588770249
```

Exercise 25

```
In[64]:= 26*26*10000+100*26^4
Out[64]= 52457600
```

Exercise 27

```
In[65]:= (26 26 + 26^3) (100 + 1000)
Out[65]= 20077200
```

Exercise 29

■ (a)

```
In[66]:= Length[CharacterRange["a", "z"]]
Out[66]= 26
```

```
In[67]:= Length[Complement[CharacterRange["a", "z"], {"a", "e", "i", "o", "u"}]]
Out[67]= 21
```

```
In[68]:= 21^8
Out[68]= 37822859361
```

■ (b)

```
In[69]:= 21 20 19 18 17 16 15 14
Out[69]= 8204716800
```

■ (c)

```
In[70]:= 5 26^7
Out[70]= 40159050880
```

■ (d)

```
In[71]:= 5 25 24 23 22 21 20 19
Out[71]= 12113640000
```

■ (e)

```
In[72]:= 26^8 - 21^8
Out[72]= 171004205215
```

■ (f)

Where is vowel? 1, 2, 3, 4, 5, 6, 7, 8

```
In[73]:= 8*5*21^7
Out[73]= 72043541640
```

■ (g)

Start with x

```
In[74]:= 26^7
Out[74]= 8031810176
```

Start with x no vowels

```
In[75]:= 21^7
Out[75]= 1801088541
```

```
In[76]:= 26^7 - 21^7
Out[76]= 6230721635
```

■ (h)

```
In[77]:= 26^6 - 21^6
Out[77]= 223149655
```

Exercise 31

■ (a)

```
In[78]:= 0
Out[78]= 0
```

■ (b)

```
In[79]:= 5!
Out[79]= 120
```

■ (c)

```
In[80]:= 6 5 4 3 2
Out[80]= 720
```

■ (d)

```
In[81]:= 7 6 5 4 3
Out[81]= 2520
```

Exercise 33

■ (a)

If n = 1, then 2.

If n = 2, then 2.

If n > 2, none

■ (b)

If n = 1, then 1.

If n = 2, then 1.

If n > 2, 2^{n-2}

■ (c)

If n = 1, then 0.

If n = 2, then 2.

If n > 2, then 2(n-1).

Exercise 37

If n is even, then 2^(n/2). If n is odd, then 2^((n+1)/2). The common answer is 2^Ceiling[n/2].

Exercise 38

■ (a)

```
In[82]:= 9*8*7*6*5*6 (* bride is in the picture *)

Out[82]= 90720

In[83]:= 9*8*7*6*5*6 (* groom is in the picture *)

Out[83]= 90720
```

■ (b)

```
In[84]:= 10*9*8*7*6*5 (* all possible *)

Out[84]= 151200

In[85]:= 8*7*6*5*4*3 (* no bride no groom *)

Out[85]= 20160

In[86]:= 10*9*8*7*6*5-8*7*6*5*4*3 (* at least one of b or g *)

Out[86]= 131040

In[87]:= 2 * 90720-131040 (* both b and g *)

Out[87]= 50400

In[88]:= 50400 (* both b and g *) + (90720-50400) (* only b *) +
          (90720-50400) (* only g *) + 20160 (* neither b or g*)

Out[88]= 151200
```

■ (c)

```
In[89]:= (* exactly one of b or g *)

In[90]:= (90720-50400) (* only b *) + (90720-50400) (* only g *)

Out[90]= 80640
```

Exercise 39

■ (a)

```
In[91]:=  2 * 5 !

Out[91]=  240
```

■ (b)

```
In[92]:=  6 ! - 2 * 5 !

Out[92]=  480
```

■ (c)

```
In[93]:=  6 ! / 2

Out[93]=  360
```

Exercise 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

```
In[94]:=  32 + 16 - 4

Out[94]=  44
```

Mathematica verification

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

```
In[95]:=  #[[1]] &[{1, 0, 0, 1, 0, 1, 1}]

Out[95]=  1
```

The following command will tell me what problem is asking for

```
In[96]:=  Or[And[#[[1]] == 0, #[[2]] == 0], And[#[[7]] == 1, #[[6]] == 1, #[[5]] == 1]] &[{1, 0, 1, 0, 0, 1, 1}]

Out[96]=  False
```

```
In[97]:= Or[And[#[[1]] == 0, #[[2]] == 0], And[#[[7]] == 1, #[[6]] == 1, #[[5]] == 1]] &[{0, 0, 1, 0, 0, 1, 1}]

Out[97]= True

In[98]:= Length[Select[Strings[{0, 1}, 7],
           Or[And[#[[1]] == 0, #[[2]] == 0], And[#[[7]] == 1, #[[6]] == 1, #[[5]] == 1]] &]]

Out[98]= 44
```

Exercise 41

```
In[99]:= 2^7 + 2^8 - 2^5

Out[99]= 352
```

Exercise 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

```
In[100]:=
        32 + 5 * 16

Out[100]=
        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

```
In[101]:=
        2 * 112 - 2
Out[101]=
        222
```

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

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

```
In[102]:=
        Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]
Out[102]=
        {{1}, {0}, {1}, {0, 0}, {1}, {0, 0, 0, 0}}
```

The following command will count how many consecutive bits there are

```
In[103]:=
        Length[#] & /@ Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]
Out[103]=
        {1, 1, 1, 2, 1, 4}
```

The following command will find the maximum number of consecutive bits

```
In[104]:=
        Max[Length[#] & /@ Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]]
Out[104]=
        4
```

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

```
In[105]:=
        Max[Length[#] & /@ Split[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]] ≥ 5
Out[105]=
        False
```

Make this into a function of a bit string

```
In[106]:=
        Max[Length[#] & /@ Split[#]] ≥ 5 &[{1, 0, 1, 0, 0, 1, 0, 0, 0, 0}]
Out[106]=
        False
```

In[107]:=
```
Length[Select[Strings[{0, 1}, 10], Max[Length[#] & /@ Split[#]] ≥ 5 &]]
```
Out[107]=
```
222
```

■ An alternative count:

The number of strings with exactly 5 consecutive 0s

In[108]:=

$$e5 = (2^4) + (2^3) + (2^3) + (2^3) + (2^3) + 2^4$$

Out[108]=
```
64
```

The number of strings with exactly 6 consecutive 0s

In[109]:=

$$e6 = (2^3) + (2^2) + (2^2) + (2^2) + 2^3$$

Out[109]=
```
28
```

The number of strings with exactly 7 consecutive 0s

In[110]:=

$$e7 = (2^2) + (2^1) + (2^1) + 2^2$$

Out[110]=
```
12
```

The number of strings with exactly 8 consecutive 0s

In[111]:=

$$e8 = (2^1) + (2^0) + (2^1)$$

Out[111]=
```
5
```

The number of strings with exactly 9 consecutive 0s

In[112]:=

$$e9 = (2^0) + (2^0)$$

Out[112]=
```
2
```

The number of strings with exactly 10 consecutive 0s

In[113]:=
```
e10 = 1
```
Out[113]=
```
1
```
The number of strings with at least 5 consecutive 0s

In[114]:=
```
e5 + e6 + e7 + e8 + e9 + e10
```
Out[114]=
```
112
```
The number of strings with at least 5 consecutive 1s is the same

The number of strings with at least 5 consecutive 0s or at least 5 consecutive 1s

In[115]:=
```
2 (e5 + e6 + e7 + e8 + e9 + e10) - 2
```
Out[115]=
```
222
```
Total

In[116]:=

2^{10}

Out[116]=
```
1024
```
In[117]:=
```
Length[Select[Strings[{0, 1}, 10], Max[Map[Length, Split[#]]] > 4 &]]
```
Out[117]=
```
222
```

Exercise 43

How many bit strings of length eight contain either at least three consecutive 0s or at least four consecutive ls?

Total

In[118]:=
```
2^8
```
Out[118]=
```
256
```
Consider the following sets of strings having exactly three consecutive 0s

S_a the set of strings with exactly three consecutive 0s which start with 0001, that is of the form 0001****. There are $2^4 - 1$ such strings, since we have to exclude 00010000.

S_b the set of strings with exactly three consecutive 0s which start with 10001, that is of the form 10001***. There are 2^3 such strings.

S_c the set of strings with exactly three consecutive 0s which are of the form *10001**. There are 2^3 such strings.

S_d the set of strings with exactly three consecutive 0s which are of the form **10001*. There are 2^3 such strings.

S_e the set of strings with exactly three consecutive 0s which end with 10001, that is are of the form ***10001. There are 2^3 such strings.

S_f the set of strings with exactly three consecutive 0s which end with 1000, that is are of the form ****1000. There are $2^4 - 1$ such strings since we have to exclide 00001000.

There is one string in $S_a \cap S_e$. There is one string in $S_a \cap S_f$. There is one string in $S_b \cap S_f$,

In[119]:=

```
ne3 = ((2^4 - 1)) + ((2^3) + (2^3) + (2^3) + (2^3)) + ((2^4 - 1)) - 3
```

Out[119]=

```
59
```

In[120]:=

```
Length[Select[Strings[{0, 1}, 8],
  (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] == 3) &]]
```

Out[120]=

```
59
```

Exactly four consecutive 0s

In[121]:=

```
ne4 = 2^3 + 2^2 + 2^2 + 2^2 + 2^3
```

Out[121]=

```
28
```

In[122]:=

```
Length[Select[Strings[{0, 1}, 8],
  (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] == 4) &]]
```

Out[122]=

```
28
```

Exactly five consecutive 0s

In[123]:=

```
ne5 = 2^2 + 2^1 + 2^1 + 2^2
```

Out[123]=

```
12
```

Exactly six consecutive 0s

In[124]:=
```
ne6 = 2 + 1 + 2
```
Out[124]=
```
5
```

Exactly seven consecutive 0s

In[125]:=
```
ne7 = 1 + 1
```
Out[125]=
```
2
```

Exactly eight consecutive 0s

In[126]:=
```
ne8 = 1
```
Out[126]=
```
1
```

In[127]:=
```
ne3 + ne4 + ne5 + ne6 + ne7 + ne8
```
Out[127]=
```
107
```

In[128]:=
```
(ne4 + ne5 + ne6 + ne7 + ne8)
```
Out[128]=
```
48
```

In[129]:=
```
(ne3 + ne4 + ne5 + ne6 + ne7 + ne8) + (ne4 + ne5 + ne6 + ne7 + ne8)
```
Out[129]=
```
155
```

Eight strings are in both:

In[130]:=
```
{0, 0, 0, 1, 1, 1, 1, 0}
```
Out[130]=
```
{0, 0, 0, 1, 1, 1, 1, 0}
```

In[131]:=
```
{0, 0, 0, 1, 1, 1, 1, 1}
```
Out[131]=
```
{0, 0, 0, 1, 1, 1, 1, 1}
```

```
In[132]:=
        {0, 0, 0, 0, 1, 1, 1, 1}
Out[132]=
        {0, 0, 0, 0, 1, 1, 1, 1}
In[133]:=
        {1, 0, 0, 0, 1, 1, 1, 1}
Out[133]=
        {1, 0, 0, 0, 1, 1, 1, 1}
In[134]:=
        {1, 1, 1, 1, 0, 0, 0, 0}
Out[134]=
        {1, 1, 1, 1, 0, 0, 0, 0}
In[135]:=
        {1, 1, 1, 1, 0, 0, 0, 1}
Out[135]=
        {1, 1, 1, 1, 0, 0, 0, 1}
In[136]:=
        {1, 1, 1, 1, 1, 0, 0, 0}
Out[136]=
        {1, 1, 1, 1, 1, 0, 0, 0}
In[137]:=
        {0, 1, 1, 1, 1, 0, 0, 0}
Out[137]=
        {0, 1, 1, 1, 1, 0, 0, 0}
```

Thus, the count of the bit strings of length eight which contain either at least three consecutive 0s or at least four consecutive ls is as follows:

```
In[138]:=
        (ne3 + ne4 + ne5 + ne6 + ne7 + ne8) + (ne4 + ne5 + ne6 + ne7 + ne8) - 8
Out[138]=
        147
```

Next I will construct *Mathematica* commands that will confirm that.

```
In[139]:=
        Map[{Length[#], First[#]} &, Split[{0, 0, 0, 1, 0}]]
Out[139]=
        {{3, 0}, {1, 1}, {1, 0}}
```

In[140]:=
```
(Max[Map[Length, Select[Split[#], First[#] == 0 &]]] > 2) &[{0, 0, 0, 1, 0}]
```
Out[140]=
```
True
```
In[141]:=
```
Length[Select[Strings[{0, 1}, 8],
  (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] > 2) &]]
```
Out[141]=
```
107
```
In[142]:=
```
Length[Select[Strings[{0, 1}, 8],
  (Max[Map[Length, Select[Split[#], First[#] == 1 &]]] > 3) &]]
```
Out[142]=
```
48
```
In[143]:=
```
Length[Select[Strings[{0, 1}, 8],
  Or[(Max[Map[Length, Select[Split[#], First[#] == 0 &]]] > 2),
    (Max[Map[Length, Select[Split[#], First[#] == 1 &]]] > 3)] &]]
```
Out[143]=
```
147
```
In[144]:=
```
Length[Select[Strings[{0, 1}, 8],
  And[(Max[Map[Length, Select[Split[#], First[#] == 0 &]]] > 2),
    (Max[Map[Length, Select[Split[#], First[#] == 1 &]]] > 3)] &]]
```
Out[144]=
```
8
```
In[145]:=
```
Select[Strings[{0, 1}, 8], And[(Max[Map[Length, Select[Split[#], First[#] == 0 &]]] > 2),
   (Max[Map[Length, Select[Split[#], First[#] == 1 &]]] > 3)] &]
```
Out[145]=
```
{{0, 0, 0, 0, 1, 1, 1, 1}, {0, 0, 0, 1, 1, 1, 1, 0},
 {0, 0, 0, 1, 1, 1, 1, 1}, {0, 1, 1, 1, 1, 0, 0, 0}, {1, 0, 0, 0, 1, 1, 1, 1},
 {1, 1, 1, 1, 0, 0, 0, 0}, {1, 1, 1, 1, 0, 0, 0, 1}, {1, 1, 1, 1, 1, 0, 0, 0}}
```

■ No consecutive 0s

In[146]:=
```
Length[Select[Strings[{0, 1}, 8],
  (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] < 2) &]]
```
Out[146]=
```
55
```

In[147]:=
```
Table[Length[Select[Strings[{0, 1}, k],
   (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] < 2) &]], {k, 1, 15}]
```
Out[147]=
```
{2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597}
```

■ At most consecutive 00s

In[148]:=
```
Length[Select[Strings[{0, 1}, 8],
  (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] < 3) &]]
```
Out[148]=
```
149
```

In[149]:=
```
Table[Length[Select[Strings[{0, 1}, k],
   (Max[Map[Length, Select[Split[#], First[#] == 0 &]]] < 3) &]], {k, 2, 15}]
```
Out[149]=
```
{4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136, 5768, 10609}
```

In[150]:=
```
Clear[sol, n];

sol[2] = 4; sol[3] = 7; sol[4] = 13;
sol[n_] := sol[n] = sol[n - 1] + sol[n - 2] + sol[n - 3]
```

In[153]:=
```
Table[sol[k], {k, 2, 15}]
```
Out[153]=
```
{4, 7, 13, 24, 44, 81, 149, 274, 504, 927, 1705, 3136, 5768, 10609}
```

In[154]:=

Solve[$x^2 - x - 1 == 0$, x]

Out[154]=

$$\left\{\left\{x \to \frac{1}{2}\left(1-\sqrt{5}\right)\right\}, \left\{x \to \frac{1}{2}\left(1+\sqrt{5}\right)\right\}\right\}$$

In[155]:=

{r1, r2, r3} = Simplify[x /. Solve[$x^3 - x^2 - x - 1 == 0$, x]]

Out[155]=

$$\left\{\frac{1}{3}\left(1+\left(19-3\sqrt{33}\right)^{1/3}+\left(19+3\sqrt{33}\right)^{1/3}\right),\right.$$
$$\frac{1}{6}\left(2+\left(-1-i\sqrt{3}\right)\left(19-3\sqrt{33}\right)^{1/3}+i\left(i+\sqrt{3}\right)\left(19+3\sqrt{33}\right)^{1/3}\right),$$
$$\left.\frac{1}{6}\left(2+i\left(i+\sqrt{3}\right)\left(19-3\sqrt{33}\right)^{1/3}+\left(-1-i\sqrt{3}\right)\left(19+3\sqrt{33}\right)^{1/3}\right)\right\}$$

In[156]:=

```
r1
```

Out[156]=

$$\frac{1}{3}\left(1+\left(19-3\sqrt{33}\right)^{1/3}+\left(19+3\sqrt{33}\right)^{1/3}\right)$$

In[157]:=

```
Simplify[Im[x /. Solve[x^3 - x^2 - x - 1 == 0, x]]]
```

Out[157]=

$$\left\{0,\ \frac{-\left(19-3\sqrt{33}\right)^{1/3}+\left(19+3\sqrt{33}\right)^{1/3}}{2\sqrt{3}},\ \frac{\left(19-3\sqrt{33}\right)^{1/3}-\left(19+3\sqrt{33}\right)^{1/3}}{2\sqrt{3}}\right\}$$

In[158]:=

```
nn = 2; Chop[NSolve[{x (r1) + y (r2) + z (r3) == 7,
   x (r1)^2 + y (r2)^2 + z (r3)^2 == 13, x (r1)^0 + y (r2)^0 + z (r3)^0 == 4}, {x, y, z}]]
```

Out[158]=

```
{{x → 3.84797, y → 0.0760144 + 0.0113183 i, z → 0.0760144 - 0.0113183 i}}
```

In[159]:=

```
Chop[Table[(x (r1)^k + y (r2)^k + z (r3)^k) /.
   {x → 3.8479711061989845`, y → 0.07601444690050753` + 0.011318308963751147` i,
    z → 0.07601444690050753` - 0.011318308963751146` i}, {k, 0, 12}]]
```

Out[159]=

```
{4., 7., 13., 24., 44., 81., 149., 274., 504., 927., 1705., 3136., 5768.}
```

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.

In[160]:=

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

In[161]:=

```
Length[allper6]
```

Out[161]=

```
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 arrangement there are two ways to arrange the bride and the groom.

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

In[162]:=

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

Out[162]=

```
True
```

In[163]:=

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

Out[163]=

```
True
```

In[164]:=

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

Out[164]=

```
240
```

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

In[165]:=

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

Out[165]=

```
192
```

In[166]:=

```
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}]] &]]
```

Out[166]=

```
192
```

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

In[167]:=

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

Out[167]=

```
{b, a}
```

In[168]:=
```
Function[y, Or[y == a, y == b]][a]
```
Out[168]=
```
True
```
In[169]:=
```
Function[y, Or[y == a, y == b]][b]
```
Out[169]=
```
True
```
In[170]:=
```
Select[#, Function[y, Or[y == a, y == b]]] &[{b, a, c, d, e, f}]
```
Out[170]=
```
{b, a}
```
In[171]:=
```
Length[Select[allper6, (Select[#, Function[y, Or[y == a, y == b]]] == {a, b}) &]]
```
Out[171]=
```
360
```