Throughout this assignment, the set \( \mathbb{N} \) of natural numbers is the set of nonnegative integers, so that in particular \( 0 \in \mathbb{N} \). The symbol \( \mathbb{N} \) can be obtained in \LaTeX{} with the command \texttt{\textbackslash naturals}. This command works in both math and text mode, and is defined for you in file \texttt{macros.sty}, which is automatically read by the \texttt{homework} class.

\textbf{\LaTeX{}}

Here is some notation that will be useful throughout this assignment. The following text in the PDF output:

Given \( k, n \in \mathbb{N} \), the following combinatorial counts hold for any \( n \)-set:

- Number of \( k \)-sequences:
  \[ n^k \]

- Number of \( k \)-permutations:
  \[
  (n)_k = \begin{cases} 
  \frac{n!}{(n-k)!} & \text{for } k \leq n \\
  0 & \text{for } k > n
  \end{cases}
  \]

- Number of \( k \)-subsets:
  \[
  \binom{n}{k} = \begin{cases} 
  \frac{n!}{k!(n-k)!} & \text{for } k \leq n \\
  0 & \text{for } k > n
  \end{cases}
  \]

- Number of subsets:
  \[ 2^n \]

can be obtained with the following \LaTeX{} input:

\begin{verbatim}
Given \( k, n \in \mathbb{N} \), the following combinatorial counts hold for any \( n \)-set:

\begin{itemize}
  \item Number of \( k \)-sequences:
    \[ n^k \]
  \item Number of \( k \)-permutations:
    \[
    (n)_k = \begin{cases} 
    \frac{n!}{(n-k)!} & \text{for } k \leq n \\
    0 & \text{for } k > n
    \end{cases}
    \]
  \item Number of \( k \)-subsets:
    \[
    \binom{n}{k} = \begin{cases} 
    \frac{n!}{k!(n-k)!} & \text{for } k \leq n \\
    0 & \text{for } k > n
    \end{cases}
    \]
  \item Number of subsets:
    \[ 2^n \]
\end{itemize}
\end{verbatim}

The \texttt{leftbrace} environment is defined for you in \texttt{macros.sty} as an \texttt{array} environment with a brace on the left. Note the ampersand (\&) to separate the two parts on each line in a \texttt{leftbrace} environment and the double slash (\textbackslash\textbackslash) to separate lines.
Part 1: Combinatorics

Find as simple a solution as possible to the problems in this part. For each problem, explain how you derive your answer in clear and succinct English and math, then give a complete formula that summarizes your explanation and write the numerical value next to it. Give an exact numerical answer, not an approximation. For instance, consider the following question:

You want to put \( h = 8 \) hens and \( r = 5 \) roosters into \( h + r = 13 \) cages arranged in a row, one animal per cage. Since roosters have a tendency to fight each other, you place them so that no two roosters are in adjacent cages. Hens are distinguishable from each other, and so are roosters, so arrangements where different hens occupy the same cages are different (and similarly for roosters). How many ways are there to place these 13 animals into the 13 cages?

Here is a sample answer:

First create a full permutation of the \( h \) hens in one of \( h! \) ways. The \( r \) roosters can then go into \( r \) of the following \( h + 1 \) positions: Just to the left of the leftmost hen, the \( h - 1 \) positions between hens, or just to the right of the rightmost hen. Each possible rooster placement is therefore a \( r \)-permutation out of an \( (h + 1) \)-set, and can be chosen in \( (h + 1) \) \( r \) ways. So the total number of animal arrangements is

\[
n = h! \times (h + 1)_r = 8! \times (9)_5 = 609,638,400
\]

for \( h = 8 \) and \( r = 5 \).

Note the general formula in terms of \( h \) and \( r \) before its numerical version for the specific problem. Also, a numerical approximation such as \( 6.1 \times 10^8 \) would not be correct. Please do use commas to separate thousands in large integers, for better readability.

To obtain exact numerical values for your answers you may want to use the Python interpreter as a calculator. The \texttt{sympy} package for symbolic computation has a reasonably efficient implementation of combinatorial formulas. To install this package, which does not come with the standard Python distribution, open a terminal (or command line window) and type

\[
\text{python3 -m pip install sympy}
\]

This should work in most cases. If not, start from this documentation page to figure out what to do.

You do not want all of \texttt{sympy} for your calculations. In IDLE, just say

\[
\text{from sympy.functions.combinatorial.numbers import nC, nP}
\]

Basic combinatorial formulas can then be computed as follows:

<table>
<thead>
<tr>
<th>Math</th>
<th>Python</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n^k )</td>
<td>( n ** k )</td>
</tr>
<tr>
<td>( (n)_k )</td>
<td>( nP(n, k) )</td>
</tr>
<tr>
<td>( \binom{n}{k} )</td>
<td>( nC(n, k) )</td>
</tr>
<tr>
<td>( 2^n )</td>
<td>( 2 ** n )</td>
</tr>
<tr>
<td>( n! )</td>
<td>( nP(n, n) )</td>
</tr>
</tbody>
</table>

For example, the numerical value for the sample answer above can be computed with the following Python code:

\[
nP(8, 8) * nP(9, 5)
\]

**General Hint:** While the problems in this part involve only simple combinatorial formulas, each requires some thinking, and the difficulty of the problems varies. In most cases, you may want to draw a diagram to see what is going on, and then work out a small example in detail. For instance, for the sample problem above, you could think of three hens and two roosters, and draw a few ways to arrange the animals into five cages. Then think of a systematic way to do so, and generalize. If you are patient, you could actually draw all 72 arrangements to verify your answer. Even better would be to write a simple piece of Python code to generate and count the “drawings” automatically (of course, just for a small example!). This is a great way to double-check your answer, as you can also test a few more results from your formula against the Python-generated drawings. The code for the hens and roosters example was worked out in class.

\textbf{Do not hand in or show any code you write for this part or the next. Thus, there is no Sakai submission for this assignment.}
Problem 1.1 (tag: one)
What is the number \( n \) of permutations \( \pi \) of \( \{1..k\} \) for \( k \geq 1 \) that satisfy \( \pi(1) \neq 1 \) (that is, the first element of the permutation is not 1)? Use \( k = 6 \) for the numerical part of your answer.

[Hint: How many permutations satisfy \( \pi(1) = 1 \)?]
[Sanity check: For \( k = 4 \) the answer is 18.]

Problem 1.2 (tag: odd)
What is the number \( n \) of subsets of the set \( S = \{1..(2k)\} \) that contain at least one odd integer, for \( k \geq 1 \)? Use \( k = 5 \) for the numerical part of your answer.

[Hint: How many subsets do not contain any odd integer?]
[Sanity check: For \( k = 2 \) the answer is \( n = 12 \).]

Problem 1.3 (tag: marriage)
There are \( m \) men and \( w \) women with \( 1 \leq w \leq m \). Each woman marries one of the men. What is the number \( n \) of ways in which this can be done? Use \( w = 4 \) and \( m = 6 \) for the numerical part of your answer.

[Hint: Choose the men, and then the couples.]
[Sanity check: For \( w = 2 \) and \( m = 3 \) the answer is 6.]

Problem 1.4 (tag: seats)
What is the number \( n \) of ways in which \( k \geq 3 \) people can be seated in a circle if two arrangements are considered the same whenever each person has the same neighbors (not necessarily on the same side)? Use \( k = 7 \) for the numerical part of your answer.

[Hint: First answer the question for the case in which the \( k \) people are seated in a row. By what factor are you over-counting when you do that, if you want the answer for people seated in a circle instead? You may want to work through a small case, say, \( k = 4 \), and then generalize.]
[Sanity check: For \( k = 5 \) the answer is 12.]

Problem 1.5 (tag: pairs)
\( 2g \) people, with \( g \geq 1 \), split up into \( g \) groups of two each. What is the number \( n \) of ways in which this can be done? Use \( g = 5 \) for the numerical part of your answer.

[Hint: Order the people in a fixed, arbitrary way. Pick a partner for person 1, and remove the chosen pair from consideration. Then choose a partner for the first of the remaining people, \ldots]
[Sanity check: For \( g = 4 \) the answer is 105.]

Problem 1.6 (tag: groups)
\( pg \) people, with \( g \geq 1 \) and \( p \geq 1 \), split up into \( g \) groups of \( p \) people each. What is the number \( n \) of ways in which this can be done? Use \( g = 5 \) and \( p = 3 \) for the numerical part of your answer.

[For a sanity check of your result, try \( p = 2 \) (previous problem), \( p = 1 \), and \( p = g \). Do not show your sanity check.]

Problem 1.7 (tag: river)
What is the number \( n \) of different ways in which the letters of the word MISSISSIPPI can be arranged if the four S’s cannot appear consecutively? No general formula is needed here, just give a numerical formula and its integer value.

Problem 1.8 (tag: bits)
Consider the sequences \( (a_1, \ldots, a_{i+j}) \) consisting of \( i \) zeros and \( j \) ones, with \( 1 \leq i \leq j + 1 \). What is the number \( n \) of such sequences if no two consecutive terms are both zeros? Use \( i = 4 \) and \( j = 8 \) for the numerical part of your answer.

[Hint: The hens and roosters problem in the introduction is different, but may give you some ideas.]
[Sanity check: For \( i = 2 \) and \( j = 5 \) the answer is 15.]
Part 2: Counting Maps

Problems in this part still involve counting, so the same instructions hold here as for part 1 of this assignment as far as answer style is concerned.

For the problems in this part, we consider the number \( n(a, b) \) of various types on maps from domain \( A = \{1..a\} \) to codomain \( B = \{1..b\} \) where \( a \) and \( b \) are positive natural numbers.

Make sure that your answers are correct for all positive natural values of \( a \) and \( b \). You may sometimes need the \texttt{leftbrace} environment described in the introduction to this assignment in order to specify values of the answer \( n(a, b) \) for different cases.

Problem 2.1 (tag: maps)

Given and explain a general formula for the number \( n(a, b) \) of maps from \( A \) to \( B \). Remember that maps cannot be empty.

Give the following values as numerical examples: \( n(3, 3) \), \( n(2, 4) \) and \( n(5, 3) \).

[Sanity check: \( n(2, 2) = 15 \).]

Problem 2.2 (tag: functions)

Given and explain a general formula for the number \( n(a, b) \) of functions from \( A \) to \( B \).

Give the following values as numerical examples: \( n(3, 3) \), \( n(2, 4) \) and \( n(5, 3) \).

[Sanity check: \( n(4, 4) = 256 \).]

Problem 2.3 (tag: injections)

Given and explain a general formula for the number \( n(a, b) \) of injections from \( A \) to \( B \).

Give the following values as numerical examples: \( n(3, 3) \), \( n(2, 4) \) and \( n(5, 3) \).

[Sanity check: \( n(4, 4) = 24 \).]

Problem 2.4 (tag: bijections)

Given and explain a general formula for the number \( n(a, b) \) of bijections from \( A \) to \( B \).

Give the following values as numerical examples: \( n(3, 3) \), \( n(2, 4) \) and \( n(5, 3) \).

[Sanity check: \( n(4, 4) = 24 \).]