Problem Set 4

Reminder:

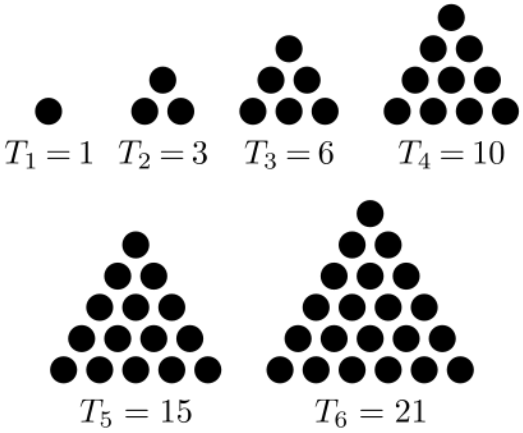
1. Name your document to `problemSet_4.R`.
2. The due day for Problem Set 4 is **October 28, 2022 2:29 pm**.
3. If you got nothing for your result, consider the following situation: 1) Any unnamed variable shown on your script; 2) Correct variable/function name for each question.
4. Only use the following packages: **`magrittr`, `readr`, `tidyr`, `dplyr`**

Warm Up

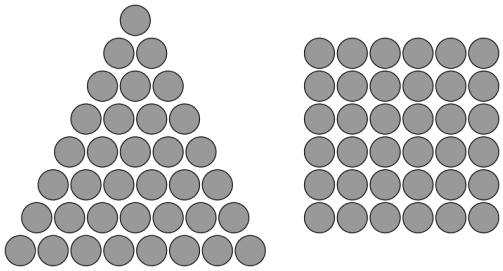
0. Enter your Name under `myName`.
1. Create a function `print_order` that given a numeric vector x with length(3), it will return the elements in order from high to low. You must use `if`, `else if`, and `else` in the function. For instance, given x1 is (1, 3.7, 6), your result should be (6, 3.7, 1).
2. Create a function `print_string` to print the numbers from 1 to any number, and print "Yes" for multiples of 3, print "No" for multiples of 5, and print "Unknown" for multiples of both. The following should be your output when number is 5:

```
## [1] 1
## [1] 2
## [1] "Yes"
## [1] 4
## [1] "No"
```

3. Create a function `calc_sum_of_factor` to calculate the sum of square of the factors of a given number. Must use **sapply** in the function. For instance the given number 12, the factors of 12 are 1, 2, 3, 4, 6, 12, your return should be 210.
4. Create a function `find_intersect` to find the intersection of three vectors. You cannot use build-in function `intersect()`.
5. Create a function `factorial_base` to calculate the factorial of a number, you cannot use build-in function `factorial()`.
6. If we want to find the sum of the first n terms, where n is a positive integer, we have the formula of Arithmetic progression: $1 + 2 + 3 + ... + n = \frac{n(n+1)}{2}$. Consider if we have a number of dots, T_n, such that $T_n = 1 + 2 + 3 + ... + n = \frac{n(n+1)}{2}$, can be arranged in a triangular form with n dots on each side.



In addition, there are some numbers of dots, like 36, can form both a triangle and a square.



Suppose you wish to find all such numbers from 1 to $T_{1500000}$. Create the following functions:
1. `T(n)`, which returns T_n for the given value of n.
2. `perfect_sqr(x)`, which returns $TRUE$ if x is a perfect square and $FALSE$ otherwise. *Hint: Use trunc(x)*
3. `num_tri_sqr(n)`, which will return all values of T_k where $1 \leq k \leq n$ and T_k is a perfect square.

What is the sum of the (only eight) T_k values that are also perfect squares, where k ranges from 1 to 1500000, Name `q6_sum`?

2022 H-1B Employer Data Hub:

On April 1, 2019, USCIS launched the H-1B Employer Data Hub to provide information on employers petitioning for H-1B workers. The data hub provides an additional layer of transparency to the H-1B program by allowing the public to search for H-1B petitioners by fiscal year, NAICS code, employer name, city, state, or ZIP code. In this section, you will explore the data in Fiscal year 2022.

Your Assignments:

1. Read the data with `read_csv()`, and Name `h1b_2022`, from the following website: https://www.uscis.gov/sites/default/files/document/data/h1b_datahubexport-2022.csv.
2. Read carefully for the data description under **USCIS**.

3. Count the **number of NA** name `na_num`, then remove all NA value and Non descriptive value(eg. State: -) from the data name `h1b_2022a`.

4. Using `h1b_2022a` to create a new dataframe that include the following contents (By order) Name by `df_num`:

   - `Init App`: The total number of initial approval application for H1b visa;
   - `Conti App`: The total number of continuing approval application for H1b visa;
   - `Approve`: Total number of Approve cases;
   - `Denial`: Total number of Denial cases.
   - The following is an example of your dataframe:

Table 1: Question 4 Dataframe

State	Initi App	Conti App	Approve	Denial
AK	12	43	12	0
AL	295	631	289	6
AP	1	0	1	0

5. Count the total number of Approval `app_num` and Denial `den_num` using `df_num`.

6. Find out the number of application in each city, it should generate a dataframe like below, and name `city_num`.

Table 2: Question 6 Dataframe

City	Count
ABBEVILLE	1
ABBOTSFORD	1
ABBOTT PARK	20

7. Find out the number of different visa applications, by using `NAICS` column, and calculate the percentage of NAICS, round 3 digits and use 100 base (0.002 * 100 = 0.2), name `visa_num`, the output should look like the following:

Table 3: Question 7 Dataframe

NAICS	Number	Percentage
11	116	0.252
21	169	0.367
22	312	0.677

Extra Bonus

You already created a function to calculate the factorial on question 5. However, this is under integer level. For this bonus question, create a function `non_integer_factorial` that can calculate the non-integer factorial, like question 5 you **cannot** use `factorial()`. Hint: Gamma Function.