

Computer Project 06

1. Assignment Overview (learning objectives)

This assignment focuses on designing, implementing, and testing a Python program using lists and dictionaries, files, sets, and tuples to solve some network-related problems.

This assignment will use the specified data structures (lists, dictionaries, files, sets, and tuples) to extract data from a network of friends. In social networks like X and Facebook, each member has friends, and the friendship relation forms a network which in mathematics is called a `graph`. The primary focus lies in identifying three individuals who are mutually connected (pairwise friends).

2. Important Note about submission

This assignment is worth 50 points (5.0% of course grade). It has 2 parts:

- Part 1 (10 Pts extra credit opportunity): a written response part (pseudocode) that must be completed before 11:59PM EST on **Tuesday, March 12th** to receive the extra credit. You need to finish Part 1 before Part 2. But you should not wait for Part 1 to be graded to do Part 2.
- Part 2: a solution in Python that must be completed before **11:59 PM EST on Monday, March 25th**.

After the due date (03/25), your score will be deducted by 10% for every 5 hours late or a fraction of it. No submissions will be accepted after 24 hours from the due date. The penalty will be applied on the full project score.

The project will be automatically submitted by the due date (03/25). If you want to keep working on it after the deadline with penalty, you must manually click on “Mark as uncompleted”, acknowledge the penalty message and manually click on “Mark as completed” when done.

3. Assignment Deliverable

The deliverable for this assignment is the following file:

`proj06.py` – the source code for your Python program

Be sure to use the specified file name and to submit it for grading via **Codio** before the project deadline.

4. Assignment Background

Social networks are integral to modern life, constituting essential elements of daily existence. Within these networks, individuals may have varying numbers of friends or followers, including the possibility of having none. Examples of social networks include Facebook, X (formerly Twitter), and Slack. Extracting information from social networks is a vibrant area of research, as businesses often align their offerings with user needs. Also, this type of analysis is used to examine criminal and terrorist organizations. Most famously, in 2022 network analysis was used to find and identify the gang members involved in governor Whitmer kidnapping plot. It was also used to find Saddam Hussein, President of Iraq, who was hiding from the coalition that invaded Iraq.



5. Assignment Specifications and Requirements

You will develop a Python program that takes 3 files (one CSV file and 2 TXT files). Check section 5.4 below for more details about the content of each file. There are 2 parts to this assignment: (1) Part 1: section 5.1 which is an extra credit opportunity that is due March 12th, and (2) a main part that is due March 25th.

5.1. Part 1: Writing a Pseudocode (Extra-credit: 10Pts)

In this task, you will have to write an algorithm for this project (section 5.2 through 5.4). An example of a pseudocode is the outline provided for the main function for project 4. You can watch the project video to get you started.

A Pseudocode is a way of representing algorithms or program logic in a structured natural language format. It is not tied to any specific programming language syntax but rather describes the steps of an algorithm in a more human-readable and understandable manner. Pseudocode is often used during the design phase of code development to outline the logic and flow of a program before actual coding begins. It helps programmers to understand the intended functionality and structure of the code without getting tangled with the specifics of a particular programming language. Pseudocode typically uses plain language mixed with programming-like constructs such as loops, conditional statements, and function calls to express the algorithm's steps.

A pseudocode can later be easily translated into a program. Pseudocode can become your code header, docstrings, and comments in part 2. When writing a pseudocode, individuals often develop their own style of presentation, recognizing that humans, rather than computers, will be reading it; thus, the rules governing pseudocode are less strict than those of a programming language. Nevertheless, adhering to some simple guidelines can enhance the universality of understanding pseudocode.

Here are a few general guidelines for checking your pseudocode:

1. Mirror effective code and clear English (avoid unnecessary full sentences).
2. Omit superfluous specifics; focus on clarity over syntax. If you are worrying about the placement of colons, you are using too many details.
3. Avoid stating the obvious; contextual clarity often suffices for variable types.
4. Utilize programming shortcuts like branching and looping for brevity.
5. Ensure a balanced level of details for readability and translatability into actual code.
6. You can start by creating the general outline of the solution: thinking about how to devise your code and what functions to use. Then, you can elaborate on each function.

Here are a couple of links on how to create such English "pseudocodes". Note that this is just an example. It is not the only way.

[https://urldefense.com/v3/_https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/_;!!HXCxUKc!matchdl3l-NUqUUSsle_SuDm_GYY329LVToMYiCNujyR4oQcSOVQYpPppVXN6c\\$](https://urldefense.com/v3/_https://www.geeksforgeeks.org/how-to-write-a-pseudo-code/_;!!HXCxUKc!matchdl3l-NUqUUSsle_SuDm_GYY329LVToMYiCNujyR4oQcSOVQYpPppVXN6c$)
[https://urldefense.com/v3/_https://www.wikihow.com/Write-Pseudocode_!!HXCxUKc!matchdl3l-NUqUUSsle_SuDm_GYY329LVToMYiCNujyR4oQcSOVQYpPJnbGwFY\\$](https://urldefense.com/v3/_https://www.wikihow.com/Write-Pseudocode_!!HXCxUKc!matchdl3l-NUqUUSsle_SuDm_GYY329LVToMYiCNujyR4oQcSOVQYpPJnbGwFY$)

5.2. General Requirements – Part 2 (50 Pts)

These are general requirements that you need to satisfy to **ensure a full score**:

1. Items 1-9 of the Coding Standard will be enforced for this project:
<http://www.cse.msu.edu/~cse231/General/coding.standard.html>
2. Per the syllabus, if you "hard code" answers, you will receive a grade of zero for the whole project.
3. You should only use materials covered in class up to week 10 (videos and textbook).
4. You must use nested dictionaries in the solution, and they must be used in a meaningful way.
 (For example, I read into a dictionary indexed by usernames where each name had a dictionary by platform with values as friends.)
5. There must be at least 10 functions that must perform non-identical operations and do/use them in a meaningful way.
 - a. There must be a function named `open_file()` that takes as argument a prompt string and returns a file pointer. If a file is not found, you must display a proper message and re-prompt.
 - b. There must be a function that reads all 3 files and *returns a nested dictionary*. You decide what are the parameters for the function. You also get to decide the organization of your nested dictionary. All files must be closed in this function and never opened again for reading. Hint: you can create helper functions to do this task.
 - c. A good practice is to have functions that perform one task at a time. It makes your code more readable.
 - d. Any function that you create needs to be used in your code.
 - e. There must be a function named `main()` (check section 5.2 for more details about the project specifications). The main function is called by:

```
if __name__ == '__main__':
    main()
```

- f. There must be *at least 7 other functions* that do something meaningful (it is good to have more functions).

5.3. Assignment Specifications

The following is the project menu. So, you may already guess what we want to do and devise your approach. You can also think what the functions are to use.

```
choices = ""
```

```
Menu :
```

- ```
1: Max number of friends intersection between X and Facebook among all
2: Percentage of people with no shared friends between X and Facebook
3: Individual information
4: Percentage of people with more friends in X compared to Facebook
5: The number of triangle friendships in X
6: The number of triangle friendships on Facebook
7: The number of triangle friendships in X and Facebook together
Enter any other key(s) to exit
```

```
""
```

### 5.3.1 Reading files

A good strategy to read the data: (you can have your own method):

Step 1- Read the first file (the names file) to a list.

Step 2- Read the second file (either Twitter or Facebook file) into a list of friends (list of lists or list of sets).

Step 3- Read the third file (either Twitter or Facebook file) into a list of friends (list of lists or list of sets).

Step 4- Then construct the nested dictionary.

### 5.3.2 Main function:

In the main function, we first need 3 inputs to get the name of the files (names, X and Facebook-related files) using the appropriate strings from the starter code or the `strings.txt` file.

Then, the main function prints the Menu (as described above) and prompts for a choice. Then,

- If the input is 1,2, 3,4,5,6, or 7:
  - It outputs the related task options as will be detailed.
  - And then, reprint the Menu.
- If the input is not one of the valid choices (any other string), it prints “Thank you” and quits the program.

Based on the Menu, below is the details of each task:

**Option 1-** For each person, we determine the intersection between their friends in X and Facebook. We identify the individual with the highest number of friends in the intersection and print the size of the largest intersection. A sample output is provided below:

```
Input a choice ~: 1
The Max number of friends in common between X and Facebook is: 1
```

**Option 2-** Calculate the percentage of individuals with no shared friends between X and Facebook and print it with the given format. This involves counting all individuals (e.g., Jack) for whom the intersection of their friends in X and Facebook is empty.

$$|\{\text{Friends of Jack in X}\} \cap \{\text{Friends of Jack in Facebook}\}| = 0$$

Then, round to the nearest integer. Here is a sample output:

```
Input a choice: 2
98% of people have no friends in common on X and Facebook
```

**Option 3-** Prompt the user to enter a name and print the X and Facebook friends associated with that name sorted in alphabetic order. The program should continue to prompt for a name until an existing name is entered.

Here is a sample output:

```
Input a choice: 3
Enter a person name: Ashely
Invalid Name or does not exist
Enter a person name: rfverg
Invalid Name or does not exist
Enter a person's Name: Katherine Olson

Friends in X

Andrea Walsh
Brady Cooper
David Sanchez
John Martin
Jordan Hopkins
Thomas Bass
Wendy Mcdaniel

Friends in Facebook

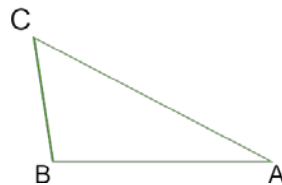
George Stewart
```

**Option 4-** Calculate the percentage of individuals who have more friends in X compared to Facebook. Round the percentage to the nearest integer.

Sample output :

```
Input a choice: 4
43% of people have more friends in X compared to Facebook
```

**Option 5-** Determine the number of triangle friendships among individuals in the X network. Persons A, B, and C are in a triangle friendship if they are pairwise friends (A is friends with B, B is friends with C, and C is friends with A).



**Warning:** A triangle friendship involving three people, A, B, and C, can appear in six different permutations: ABC, ACB, CBA, CAB, BCA, and BAC. All six permutations are counted as one triangle friendship.

A raw but inefficient strategy involves using three nested loops to iterate over every combination of three people. We ensure that P1, P2, and P3 are pairwise different ( $P1 \neq P2$ ,  $P2 \neq P3$ ,  $P1 \neq P3$ ). Then we check for triangle friendships, noting that different permutations of the same set should be counted as one. More, precisely,

```
For P1 in Collection
 For P2 in Collection
 For P3 in Collection
```

Make sure that P1, P2, P3 are pairwise different ( $P1 \neq P2$ ,  $P2 \neq P3$ ,  $P1 \neq P3$ ). Then check for the triangle friendships and specify that the following should be counted as one.

```
(P1,P2,P3)
(P1,P3,P2)
(P2,P3,P1)
(P2,P1,P3)
(P3,P2,P1)
(P3,P1,P2)
```

**Warning:** python does not handle set of sets!

Sample output

```
Input a choice: 5
The number of triangle friendships in X is: 2
```

**Option 6-** Similar to option 5 except this is for the Facebook network.

Sample output

```
Input a choice: 6
The number of triangle friendships in Facebook is: 1
```

**Option 7-** Similar to 6 except this is for the merged network between Facebook and X.

Hint: Depending on your nested dictionary structure, you need to extract and use the following data for each person to find the triangle of friendships:

$$\{X's \text{ friends for Person } 1\} \cup \{\text{Facebook's friends for Person } 1\}$$

Sample output:

```
Input a choice: 7
The number of triangle friendships in X merged with Facebook is: 12
```

## 5.4. Files Description

### 5.4.1 CSV File

The **first file** is **Names.csv** which contains some names:

1. Contains 200 lines, representing a unique full name at each line.
2. Assume, line numbers start from zero.
3. Then assume, the person's ID corresponds to the line number of their name in the file.

| Name                  | ID    |
|-----------------------|-------|
| Gregory Williams      | → 0   |
| George Stewart        | → 1   |
| Mark Mercado.         | → 2   |
| Devin Baldwin         | → 3   |
| Samantha Murray       | → 4   |
| Brandon Garcia        | → 5   |
| Laura Todd            | → 6   |
| Alexander Rogers      |       |
| Michael Walter        |       |
| Michael Stewart       |       |
| Daniel Robinson       |       |
| Elizabeth Russell     |       |
| Janet Myers           |       |
| Ms. Caitlin Graham MD |       |
| .....                 |       |
| .....                 | → 199 |

### 5.4.2 Twitter (X Platform) File

The **second file**, named **twf\_Friends\_id.txt**, specifies the list of the IDs of the friends from X (Twitter).

1. Each line shows the ID of some people.
2. The actual Name of the corresponding ID can be found in the first file above, as mentioned.
3. The IDs in the *i*'th line are the friend IDs of the person in *i*'th line of the first file above.

The corresponding person does not have a friend if a line is blank.

For example, the sixth line below is blank, and thus, Brandon Garcia above does not have any friends in X.

#### twf\_Friends\_id.txt

```
32,47,
61,128,150,170,
26,
66,117,152,193,199,
159,

22,145,
61,64,115,
40,73,160,

59,71,

....
```

### 5.4.3 Facebook File

The **third file**, named **fb\_Friends.txt**, is the friend's Name from Facebook.

1. In each line, you see some names.
2. The names in the *i*'th line are the friends of a person in the *i*'th line of the first file.

Again, the corresponding person does not have a friend if a line is blank.

#### fb\_Friends.txt

```
John Moss,
Katherine Olson,

Deborah Simmons,Eric Robinson,Tammie Gallegos,
Erika Salazar,John Moss,Tiffany Avery,Patricia Henson,
Deborah Simmons,Richard Green,

Jeffrey Moon,Tyler Davis,
Charles Walker,
Marie Mack,
Brady Gomez,
Kenneth Miller DDS,Cynthia Ellison,
Martha Farmer,Jeremy Ingram,
Martha Farmer,Jonathon Esparza,

.....
```

### 5.4.4 Final Remarks about the files

As you see, the X friends are in the ID form, and the Facebook friends are in the actual name form. We need to turn the ID into actual names to uniform your data structure.

## 6. Grading Rubric

### Computer Project #06 Scoring Summary

#### General Requirements:

- ( 5 pts) Coding Standard 1-9
  - (descriptive comments, function headers, mnemonic identifiers, format, etc...)

#### Implementation:

Visible tests files (Test 1): Names.csv, twt\_Friends\_id.txt, fb\_Friends.txt

- ( 1 pt ) Check 1 (option 1)
- ( 1 pt ) Check 2 (option 2)
- ( 1 pt ) Check 3 (option 3)
- ( 1 pt ) Check 4 (option 4)
- ( 1 pt ) Check 5 (option 5)
- ( 1 pt ) Check 6 (option 6)
- ( 1 pt ) Check 7 (option 7)
- ( 1 pt ) Check 8 (Errors)

Visible tests files (Test 2): Names.csv, twt\_Friends\_id\_1.txt, fb\_Friends\_1.txt

- ( 1 pt ) Check 1 (option 1)
- ( 1 pt ) Check 2 (option 2)
- ( 1 pt ) Check 3 (option 3)
- ( 1 pt ) Check 4 (option 4)
- ( 1 pt ) Check 5 (option 5)
- ( 1 pt ) Check 6 (option 6)
- ( 1 pt ) Check 7 (option 7)
- ( 1 pt ) Check 8 (Errors)

-----  
Hidden tests with different files

- ( 1 pt ) Check 1 (option 1)
- ( 1 pt ) Check 2 (option 2)
- ( 1 pt ) Check 3 (option 3)
- ( 1 pt ) Check 4 (option 4)
- ( 1 pt ) Check 5 (option 5)
- ( 1 pt ) Check 6 (option 6)
- ( 1 pt ) Check 7 (option 7)
- ( 1 pt ) Check 8 (Errors)

-----  
Function tests (no automatic tests)

- ( 2 pts) open\_file (Manual Grading)
  - 1 point if No try/except
  - 1 point if No while loop
- ( 5 pts) function to read all files
- ( 14 pts) at least 7 functions(2 pts each)

#### Note:

- hard coding an answer earns zero points for the whole project.
- -10 points for not putting all your user interaction in your main().
- Use of any advanced data structures not covered in class earns zero points for the whole project.
- If you define a function and you do not use it then you will lose 2 Pts for each.



## 7. Sample Output:

Because the interactions are lengthy, two have been saved in separate files named `inst_out?.txt` (where ? represents a digit 1, or 2) available in the project folder. Each output file corresponds to an `input?.txt` file. You can also use the `run_file.py` to generate similar output text files to test your code. You can later copy the content of both files and use <https://www.diffchecker.com/> to compare the files.

Note that running the `run_file.py` with no errors does not mean you passed the test. You still need to check your output against the instructor's output.

Also, to run that file or when submitting to Codio, you need to uncomment the following lines from your starter code. Otherwise, you will fail the test. This function will echo the input to your output:

```
import sys
def input(prompt=None):
 if prompt != None:
 print(prompt, end="")
 aaa_str = sys.stdin.readline()
 aaa_str = aaa_str.rstrip("\n")
 print(aaa_str)
 return aaa_str
```

## Test 1

```
Enter a names file ~:names
Error. File does not exist
```

```
Enter a names file ~:Names.csv
```

```
Enter the twitter id file ~:friends
Error. File does not exist
```

```
Enter the twitter id file ~:twf_Friends_id.txt
```

```
Enter the facebook id file ~:facebook
Error. File does not exist
```

```
Enter the facebook id file ~:fb_Friends.txt
```

```
Menu :
```

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

```
Input a choice ~:1
```

```
The Max number intersection of friends between X and Facebook is: 1
```

```
Menu :
```

- 1: Max number of friends intersection between X and Facebook among all
- 2: Percentage of people with no shared friends between X and Facebook

- 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:2

98% of people have no friends in common on X and Facebook

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:3

Enter a person's name ~:Ashely

Invalid name or does not exist

Enter a person's name ~:rfverg

Invalid name or does not exist

Enter a person's name ~:Katherine Olson

-----  
Friends in X

\*\*\*\*\*

Andrea Walsh  
Brady Cooper  
David Sanchez  
John Martin  
Jordan Hopkins  
Thomas Bass  
Wendy Mcdaniel

-----  
Friends in Facebook

\*\*\*\*\*

George Stewart  
Roger Berger

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:4

43% of people have more friends in X compared to Facebook

Menu :

- 1: Max number of friends intersection between X and Facebook among all
- 2: Percentage of people with no shared friends between X and Facebook
- 3: Individual information

- 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:5

The number of triangle friendships in X is: 2

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:6

The number of triangle friendships in Facebook is: 1

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:7

The number of triangle friendships in X merged with Facebook is: 12

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:

Thank you

## Test 2

Enter a names file ~:names

Error. File does not exist

Enter a names file ~:Names.csv

Enter the twitter id file ~:friends

Error. File does not exist

Enter the twitter id file ~:twf\_Friends\_id\_1.txt

Enter the facebook id file ~:fb\_Friends\_1.txt

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:1

The Max number intersection of friends between X and Facebook is: 2

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:2

72% of people have no friends in common on X and Facebook

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:3

Enter a person's name ~:Ricjjhard Horn

Invalid name or does not exist

Enter a person's name ~:Richjhggard Horn

Invalid name or does not exist

Enter a person's name ~:Richard Horn

-----  
Friends in X  
\*\*\*\*\*

Christopher Allen  
Dr. Stephanie Hall MD  
Eric Fletcher  
Erika Salazar  
Ethan Aguirre  
Jeffrey Moon  
Mark Mercado  
Michael Park

Patricia Henson

Tiffany Avery

-----  
Friends in Facebook

\*\*\*\*\*

Jeremy Burke

Richard Green

Samantha Murray

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:4

75% of people have more friends in X compared to Facebook

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:5

The number of triangle friendships in X is: 135

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:6

The number of triangle friendships in Facebook is: 55

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:7

The number of triangle friendships in X merged with Facebook is: 665

Menu :

- 1: Max number of friends intersection between X and Facebook among all
  - 2: Percentage of people with no shared friends between X and Facebook
  - 3: Individual information
  - 4: Percentage of people with more friends in X compared to Facebook
  - 5: The number of triangle friendships in X
  - 6: The number of triangle friendships on Facebook
  - 7: The number of triangle friendships in X and Facebook together
- Enter any other key(s) to exit

Input a choice ~:dd

Thank you