




**University of
Nottingham**
UK | CHINA | MALAYSIA

COMP1028 Programming and Algorithm

Session: Autumn 2024

Group Coursework

Group Name	3C1M					
Group Members						
Name 1 (Leader)	Imtiaz Naufal bin Mohammad Fuad			ID 1	20612400	
Name 2	Ivan Char Cheng Jun			ID 2	20614522	
Name 3	Cham Jin Jie			ID 3	20611325	
Name 4	Ng Yoong Shen			ID 4	20609660	
Name 5				ID 5		
Marks	Program functionality and code quality (50)	Usability (10)	Comments(5)	Documentation (5)	Presentation & Demo (Recording) (15)	Bonus Feature (15)
Total (100)						
Leader Sign						
Date	8th December 2024					

Contact Management System using C

1.0 Introduction

The Contact Management System is a C-based application designed for effective management of personal or professional contact lists. Its purpose is to provide users with a reliable platform to organize and interact with contact information efficiently.

2.0 Installation

Provide step-by-step instructions on how to install and set up the project. Include any dependencies that need to be installed. Consider the possibility that whoever is reading your README is a novice and would like more guidance. Listing specific steps helps remove ambiguity and gets people to using your project as quickly as possible. Say also if it only runs in a specific context like a particular programming language version or operating system or has dependencies that have to be installed manually.

Local Installation

NOTE: Ensure that C Compiler - GCC version **9.3.0 or higher** (recommended: GCC 11.0+) is installed in your machine before running the program

Installation of GCC

To verify GCC is installed and check its version, run this command in your workplace's terminal

```
gcc --version
```

If it's not installed:

- Windows: Install [MinGW](#)
- Linux: Use your package manager to install

```
sudo apt update && sudo apt install build-essential
```

- macOS: Install Xcode Command Line Tools and uses clang to compile the code

```
xcode-select -install
```

Cloning the repository

Clone the repository

```
git clone https://github.com/imtiaznn/C-Contact-Manager.git
```

Change directory

```
cd C-Contact-Manager
```

Compiling the source files using GCC

Compile the program

```
gcc main.c file_io.c utils.c -o app.exe
```

3.0 Usage

Run the program

```
./app.exe
```

4.0 Features

List the main features of your program. This could include:

- Feature 1 - Save, edit, and delete contact from CSV file.
- Feature 2 - Search the contact from the CSV file.
- Feature 3 - Sort the contact by name, phone number or email address.

Depending on what you are making, it can be a good idea to include screenshots

1. Add Contact

- ✓ Allows users to seamlessly add new contacts with a name, phone number, and email address.
- ✓ Implements robust input validation to ensure phone numbers and email addresses adhere to correct formats.
- ✓ Prevents duplicate contact names, maintaining data uniqueness.

```
----- Contact List -----
NAME                PHONE NUMBER        EMAIL
1 - Aiman Malik bin Ridzwan Woo    +601125868682    aimanmalik@gmail.com
2 - Bob Brown                     +601144556666    bobbrown@example.com
3 - Charlie Davis                  +601155667777    charliedavis@example.com
4 - Ethan Foster                   +601177889999    ethanfoster@example.com
5 - Fiona Green                    +601188998888    fionagreen@example.com

There are 19 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>
----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
-----
? 1
(1/3) Enter the name of the contact to be saved
? Ivan Char
(2/3) Enter the Phone Number of the contact to be saved
? +601128338647
(3/3) Enter the email of the contact to be saved
? ivanchar@gmail.com
Saving Contact...
----- Contact List -----
NAME                PHONE NUMBER        EMAIL
1 - Aiman Malik bin Ridzwan Woo    +601125868682    aimanmalik@gmail.com
2 - Bob Brown                     +601144556666    bobbrown@example.com
3 - Charlie Davis                  +601155667777    charliedavis@example.com
4 - Ethan Foster                   +601177889999    ethanfoster@example.com
5 - Fiona Green                    +601188998888    fionagreen@example.com

There are 28 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>
(Success) Contact saved successfully
----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
-----
```

2. Edit Contact

- ✓ Enables users to update an existing contact's name, phone number, or email address.
- ✓ Validates input during the editing process to ensure consistency and accuracy.
- ✓ Ensures no duplicate names are introduced when making changes.

```
----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit
-----
? 2
(1/4) Enter the name of the contact to be edited
? Ivan Char

Editing contact Ivan Char...

Contact Information
Name: Ivan Char
Phone Number: +601128338647
Email: ivanchar@gmail.com

(2/4) Which value would you like to edit? Enter the index of the selected option (Press 1, 2 or 3):
1 - Name
2 - Phone Number
3 - Email
? 1
(3/4) Enter the new value
? VanVan

(4/4) Are you sure you want to edit the contact with the following new information (Y/N)?
Name: VanVan
This process is NOT reversible.
? Y
Contact 'Ivan Char' found. Performing selected operation...
(editContact) Contact successfully updated.
-----
NAME                PHONE NUMBER      EMAIL
-----
1 - Aiman Malik bin Ridzwan Woo    +601125868682    aimanmalik@gmail.com
2 - Bob Brown                     +60114455666     bobbrown@example.com
3 - Charlie Davis                  +60115566777     charliedavis@example.com
4 - Ethan Foster                   +60117788999     ethanfoster@example.com
5 - Fiona Green                    +60118899000     fionagreen@example.com

There are 20 contacts found.
PAGE (1/4)

<< (5) Previous Page                (6) Next Page >>

(Success) Contact edited successfully

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit
-----
```

3. Delete Contact

- ✓ Provides an option to remove a contact by specifying their name.
- ✓ Includes a confirmation step to prevent accidental deletions.

```
----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit
-----
? 3
(1/2) Enter the name of the contact to be deleted
? VanVan
(2/2) Are you sure you want to delete the node (Y/N)

Name: VanVan
Phone Number: +601128338647
Email: ivanchar@gmail.com

This process is NOT reversible
? Y
Contact 'VanVan' found. Performing selected operation...
-----
NAME                PHONE NUMBER      EMAIL
-----
1 - Aiman Malik bin Ridzwan Woo    +601125868682    aimanmalik@gmail.com
2 - Bob Brown                     +60114455666     bobbrown@example.com
3 - Charlie Davis                  +60115566777     charliedavis@example.com
4 - Ethan Foster                   +60117788999     ethanfoster@example.com
5 - Fiona Green                    +60118899000     fionagreen@example.com

There are 19 contacts found.
PAGE (1/4)

<< (5) Previous Page                (6) Next Page >>

(Success) Contact deleted successfully

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit
-----
```

4. Search Contact

- ✓ Supports searching for contacts using a name, phone number, or email address.
- ✓ Features partial matching for flexible and efficient searches.

```
----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice

Press '0' to Exit
-----
? 4
Enter search query (Enter a blank line to reset query):
? Bob Brown
----- Contact List -----
NAME                                PHONE NUMBER    EMAIL
-----
2 - Bob Brown                      +60114455666    bobbrown@example.com

There are 1 contacts found.

PAGE (1/1)

<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice

Press '0' to Exit
-----
```

5. Display Contacts

- ✓ Displays the contact list in a user-friendly, paginated format.
- ✓ Offers easy navigation between pages for improved accessibility.

```
----- Contact List -----
NAME                                PHONE NUMBER    EMAIL
-----
1 - Aiman Malik bin Ridzwan Woo    +601125868682  aimanmalik@gmail.com
2 - Bob Brown                      +60114455666    bobbrown@example.com
3 - Charlie Davis                  +60115566777    charliedavis@example.com
4 - Ethan Foster                   +60117788999    ethanfoster@example.com
5 - Fiona Green                   +60118899000    fionagreen@example.com

There are 19 contacts found.

PAGE (1/4)

<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice

Press '0' to Exit
-----
? 6
----- Contact List -----
NAME                                PHONE NUMBER    EMAIL
-----
6 - George Harris                  +60119908111    georgeharris@example.com
7 - Hannah Irving                  +60120811222    hannahirving@example.com
8 - Ian Jackson                    +60121112333    ianjackson@example.com
9 - Jane Smith                     +60198765432    janesmith@example.com
10 - John Doe                      +60123456789    johndoe@example.com

There are 19 contacts found.

PAGE (2/4)

<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice

Press '0' to Exit
-----
? 6
----- Contact List -----
NAME                                PHONE NUMBER    EMAIL
-----
11 - Julia King                    +60122233444    juliaking@example.com
12 - Kevin Lee                     +60123344555    kevinlee@example.com
13 - Laura Martin                  +60124455666    lauramartin@example.com
14 - Michael Nelson                +60125566777    michaelnelson@example.com
15 - Natalie Owens                 +60126677888    natalieowens@example.com

There are 19 contacts found.

PAGE (3/4)

<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice

Press '0' to Exit
-----
? |
```

6. Sort Contacts

- ✓ Allows sorting of contacts by name, phone number, or email address for better organization.

```
----- Contact List -----
NAME                                PHONE NUMBER                    EMAIL
-----
1 - Aiman Malik bin Ridzwan Woo    +601325868682                  aimanmalik@gmail.com
2 - Bob Brown                     +60124455666                   bobbrown@example.com
3 - Charlie Davis                  +60115566777                   charliedavis@example.com
4 - Ethan Foster                   +60117788999                   ethanfooster@example.com
5 - Fiona Green                    +60148899000                   fionagreen@example.com

There are 19 contacts found.

PAGE (1/4)

<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search

7.Sort by Choice

Press '0' to Exit

? 7
Enter the index of the selected option (Press 0, 1 or 2):
0. Sort by Name
1. Sort by Phone Number
2. Sort by Email
? 1
----- Contact List -----
NAME                                PHONE NUMBER                    EMAIL
-----
1 - Hannah Irving                 +60110011222                   hannahirving@example.com
2 - Charlie Davis                  +60115566777                   charliedavis@example.com
3 - Ethan Foster                   +60117788999                   ethanfooster@example.com
4 - Quincy Roberts                 +60110900111                   quincyroberts@example.com
5 - John Doe                      +60123456789                   johndoe@example.com

There are 19 contacts found.

PAGE (1/4)

<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search

7.Sort by Choice

Press '0' to Exit
```

7. Save Contacts:

- ✓ Saves all contact information into a CSV file for persistence.
- ✓ Appends new entries to the file without overwriting existing data.

Before:

```
1 Aiman Malik bin Ridzwan Woo,+601325868682,aimanmalik@gmail.com
2 Bob Brown,+60124455666,bobbrown@example.com
3 Charlie Davis,+60115566777,charliedavis@example.com
4 Ethan Foster,+60117788999,ethanfooster@example.com
5 Fiona Green,+60148899000,fionagreen@example.com
6 George Harris,+60129900111,georgeharris@example.com
7 Hannah Irving,+60110011222,hannahirving@example.com
8 Ian Jackson,+60181122333,ianjackson@example.com
9 Jane Smith,+60198765432,janesmith@example.com
10 John Doe,+60123456789,johndoe@example.com
11 Julia King,+60132233444,juliaking@example.com
12 Kevin Lee,+60163344555,kevinlee@example.com
13 Laura Martin,+60174455666,lauramartin@example.com
14 Michael Nelson,+60125566777,michaelnelson@example.com
15 Natalie Owens,+60126677888,natalieowens@example.com
16 Oliver Parker,+60187788999,oliverparker@example.com
17 Paula Quinn,+60188899000,paulaquinn@example.com
18 Quincy Roberts,+60119900111,quincyroberts@example.com
19 Rachel Scott,+60130011222,rachelscott@example.com
```

After:

```
1 Aiman Malik bin Ridzwan Woo,+601325868682,aimanmalik@gmail.com
2 Bob Brown,+60124455666,bobbrown@example.com
3 Charlie Davis,+60115566777,charliedavis@example.com
4 Ethan Foster,+60117788999,ethanfoster@example.com
5 Fiona Green,+60148899000,fionagreen@example.com
6 George Harris,+60129900111,georgeharris@example.com
7 Hannah Irving,+60110011222,hannahirving@example.com
8 Ian Jackson,+60181122333,ianjackson@example.com
9 Jane Smith,+60198765432,janesmith@example.com
10 John Doe,+60123456789,johndoe@example.com
11 Julia King,+60132233444,juliaking@example.com
12 Kevin Lee,+60163344555,kevinlee@example.com
13 Laura Martin,+60174455666,lauramartin@example.com
14 Michael Nelson,+60125566777,michaelnelson@example.com
15 Natalie Owens,+60126677888,natalieowens@example.com
16 Oliver Parker,+60187788999,oliverparker@example.com
17 Paula Quinn,+60188899000,paulaquinn@example.com
18 Quincy Roberts,+60119900111,quincyroberts@example.com
19 Rachel Scott,+60130011222,rachelscott@example.com
20 Cham TGM,+601234567893,chamjj@gmail.com
```

8. Load Contacts
 - ✓ Automatically imports contacts from a CSV file into a Binary Search Tree (BST) upon program startup.
9. Update CSV
 - ✓ Keeps the CSV file up to date with any changes, such as additions, edits, or deletions.

Before:

```
1 Aiman Malik bin Ridzwan Woo,+601325868682,aimanmalik@gmail.com
2 Bob Brown,+60124455666,bobbrown@example.com
3 Charlie Davis,+60115566777,charliedavis@example.com
4 Ethan Foster,+60117788999,ethanfoster@example.com
5 Fiona Green,+60148899000,fionagreen@example.com
6 George Harris,+60129900111,georgeharris@example.com
7 Hannah Irving,+60110011222,hannahirving@example.com
8 Ian Jackson,+60181122333,ianjackson@example.com
9 Jane Smith,+60198765432,janesmith@example.com
10 John Doe,+60123456789,johndoe@example.com
11 Julia King,+60132233444,juliaking@example.com
12 Kevin Lee,+60163344555,kevinlee@example.com
13 Laura Martin,+60174455666,lauramartin@example.com
14 Michael Nelson,+60125566777,michaelnelson@example.com
15 Natalie Owens,+60126677888,natalieowens@example.com
16 Oliver Parker,+60187788999,oliverparker@example.com
17 Paula Quinn,+60188899000,paulaquinn@example.com
18 Quincy Roberts,+60119900111,quincyroberts@example.com
19 Rachel Scott,+60130011222,rachelscott@example.com
20 Cham TGM,+601234567893,chamjj@gmail.com
```

After:

```
1 Aiman Malik bin Ridzwan Woo,+601325868682,aimanmalik@gmail.com
2 Bob Brown,+60124455666,bobbrown@example.com
3 Chamj,+601234567893,chamjinjie@gmail.com
4 Charlie Davis,+60115566777,charliedavis@example.com
5 Ethan Foster,+60117788999,ethanfoster@example.com
6 Fiona Green,+60148899000,fionagreen@example.com
7 George Harris,+60129900111,georgeharris@example.com
8 Hannah Irving,+60110011222,hannahirving@example.com
9 Ian Jackson,+60181122333,ianjackson@example.com
10 Jane Smith,+60198765432,janesmith@example.com
11 John Doe,+60123456789,johndoe@example.com
12 Julia King,+60132233444,juliaking@example.com
13 Kevin Lee,+60163344555,kevinlee@example.com
14 Laura Martin,+60174455666,lauramartin@example.com
15 Michael Nelson,+60125566777,michaelnelson@example.com
16 Natalie Owens,+60126677888,natalieowens@example.com
17 Oliver Parker,+60187788999,oliverparker@example.com
18 Paula Quinn,+60188899000,paulaquinn@example.com
19 Quincy Roberts,+60119900111,quincyroberts@example.com
20 Rachel Scott,+60130011222,rachelscott@example.com
```

10. Input Validation

- ✓ Validates phone numbers to ensure compliance with specific formats (e.g., Malaysian phone standards).

```
----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit

? 1
(1/3) Enter the name of the contact to be saved
? Ivan Char
(2/3) Enter the the Phone Number of the contact to be saved
? 12345678
----- Contact List -----
NAME                PHONE NUMBER      EMAIL
1 - Aiman Malik bin Ridzwan Woo      +601325868682    aimanmalik@gmail.com
2 - Bob Brown                       +60124455666     bobbrown@example.com
3 - Chamj                           +601234567893    chamjinjie@gmail.com
4 - Charlie Davis                    +60115566777     charliedavis@example.com
5 - Ethan Foster                     +60117788999     ethanfoster@example.com

There are 20 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>
(Error) Invalid phone number, please try again

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit
```

- ✓ Ensures email addresses are correctly formatted to adhere to standard email conventions.

```
----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit

? 1
(1/3) Enter the name of the contact to be saved
? Ivan Char
(2/3) Enter the the Phone Number of the contact to be saved
? +601123338647
(3/3) Enter the email of the contact to be saved
? ivanchar gmail.com
----- Contact List -----
NAME                PHONE NUMBER      EMAIL
1 - Aiman Malik bin Ridzwan Woo      +601325868682    aimanmalik@gmail.com
2 - Bob Brown                       +60124455666     bobbrown@example.com
3 - Chamj                           +601234567893    chamjinjie@gmail.com
4 - Charlie Davis                    +60115566777     charliedavis@example.com
5 - Ethan Foster                     +60117788999     ethanfoster@example.com

There are 20 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>
(Error) Invalid email address, please try again

----- Available Operations -----
1.Save      2.Edit
3.Delete    4.Search

7.Sort by Choice
Press '0' to Exit
```


- ✓ Ensures name is not duplicate when creating a new contact.

```
----- Contact List -----
NAME                PHONE NUMBER        EMAIL
1 - Aiman Malik bin Ridzwan Woo +601325868682      aimanmalik@gmail.com
2 - Bob Brown             +601244556666      bobbrown@example.com
3 - Chamj                 +601234567893      chamjinjie@gmail.com
4 - Charlie Davis          +601155667777      charliedavis@example.com
5 - Ethan Foster           +601177889999      ethanfooster@example.com

There are 20 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
=====
? 1
(1/1) Enter the name of the contact to be saved
? Chamj
----- Contact List -----
NAME                PHONE NUMBER        EMAIL
1 - Aiman Malik bin Ridzwan Woo +601325868682      aimanmalik@gmail.com
2 - Bob Brown             +601244556666      bobbrown@example.com
3 - Chamj                 +601234567893      chamjinjie@gmail.com
4 - Charlie Davis          +601155667777      charliedavis@example.com
5 - Ethan Foster           +601177889999      ethanfooster@example.com

There are 20 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>

(Error) Duplicate values are not allowed

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
=====
```

- ✓ Ensure search for a contact that have created before in the BST.

```
----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
=====
? 4
Enter search query (Enter a blank line to reset query):
? IvanChar
----- Contact List -----
NAME                PHONE NUMBER        EMAIL

No contacts found matching 'IvanChar'...
PAGE (1/0)
<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
=====
```

- ✓ Ensure enter the options that are listed in the program.

```
----- Contact List -----
NAME                PHONE NUMBER        EMAIL
1 - Aiman Malik bin Ridzwan Woo +601325868682      aimanmalik@gmail.com
2 - Bob Brown             +601244556666      bobbrown@example.com
3 - Chamj                 +601234567893      chamjinjie@gmail.com
4 - Charlie Davis          +601155667777      charliedavis@example.com
5 - Ethan Foster           +601177889999      ethanfooster@example.com

There are 20 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
=====
? 10
----- Contact List -----
NAME                PHONE NUMBER        EMAIL
1 - Aiman Malik bin Ridzwan Woo +601325868682      aimanmalik@gmail.com
2 - Bob Brown             +601244556666      bobbrown@example.com
3 - Chamj                 +601234567893      chamjinjie@gmail.com
4 - Charlie Davis          +601155667777      charliedavis@example.com
5 - Ethan Foster           +601177889999      ethanfooster@example.com

There are 20 contacts found.
PAGE (1/4)
<< (5) Previous Page                (6) Next Page >>

(Error) Invalid option. Please try again

----- Available Operations -----
1.Save          2.Edit
3.Delete        4.Search
7.Sort by Choice
Press '0' to Exit
=====
```

11. Utilizes dynamic memory allocation for efficient storage of contact data

- ✓ Ensures all allocated memory is properly freed, preventing memory leaks and ensuring optimal performance.

5.0 License

MIT License

Copyright (c) 2024 C-Contact-Manager

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.

6.0 Contact

For question, feedback or contribution, feel free to contact any of the developer below:

1. Imtiaz Naufal bin Mohammad Fuad

Email: imtyaz.n96@gmail.com

Github: <https://github.com/imtiaznn>

2. Ivan Char Cheng Jun

Email: ivanccj0120@gmail.com

Github: <https://github.com/VanVan120>

3. Cham Jin Jie

Email: jcham240@gmail.com

Github: <https://github.com/chamtgm>

4. Ng Yoong Shen

Email: ngyoongshen331@gmail.com

Github: <https://github.com/sunlovefools>