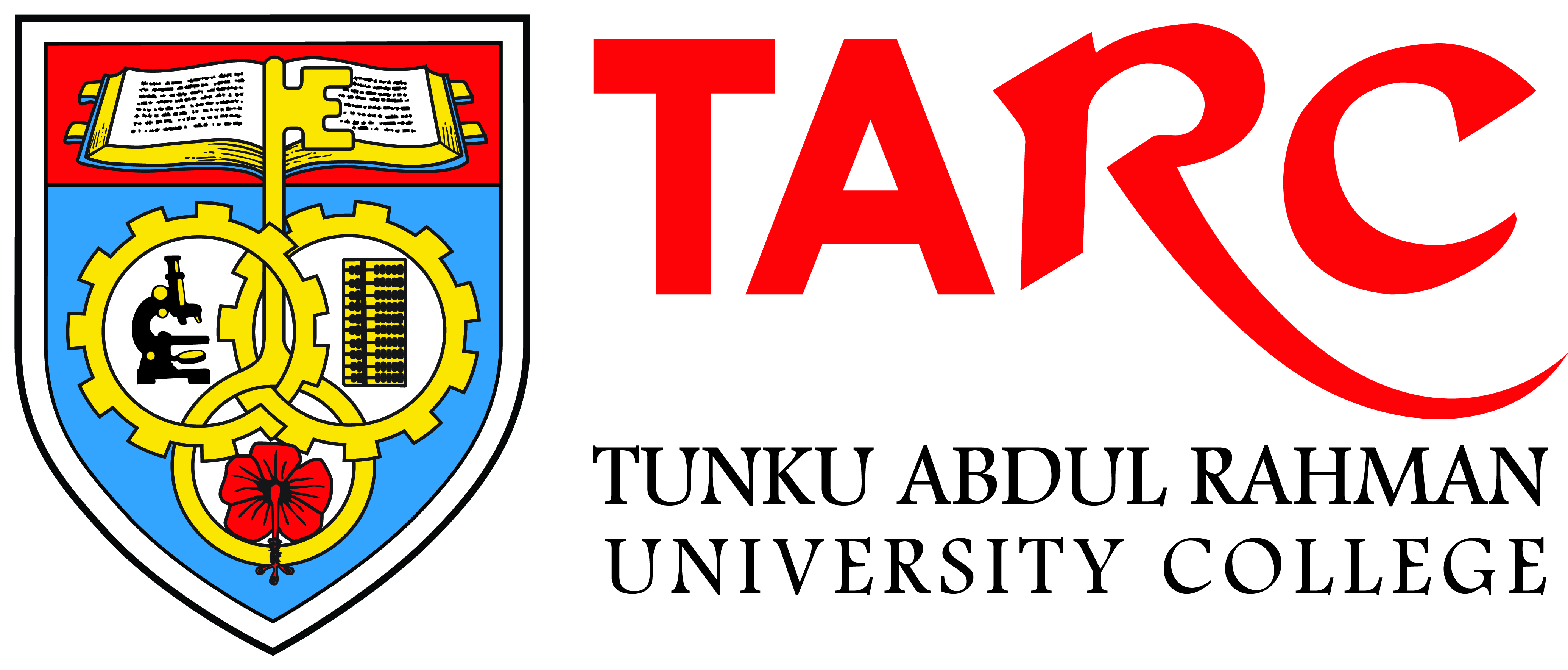
****

TUNKU ABDUL RAHMAN UNIVERSITY COLLEGE

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY

SEMESTER 1 - ACADEMIC YEAR 2020/2021

**AACS1074**

**PROGRAMMING CONCEPTS & DESIGN I**

**(ASSIGNMENT)**

**STUDENT NAME : Poh Choo Han**

**STUDENT ID : 20PMD00627**

**PROGRAMME :**  **DCO**

**TUTORIAL GROUP :**  **GROUP 1**

**TUTOR :** **MS Kheoh Hooi Leng**

**DATE RECEIVED :**

**Assessment Criteria & Feedback Form** AACS1074 PCD I Assignment

|  |  |  |  |
| --- | --- | --- | --- |
| **Program (75%)** |  | **Report (25%)** |  |
| A - Submitted C Code, shows understanding **( 5)** |  | A - Submitted Var & Const, f/chart OR **(5)**  Pseudocode, screenshots |  |
| B - Submitted C Code, shows understanding **( 5)** |  | B - Submitted Var & Const, f/chart OR **(5)**  Pseudocode, screenshots |  |
| 1. Basic calculations for different details **(15)**    * Correct inputting values    * Correct processing / calculation    * Correct output values / totals    * Correct output / report I format    * Used at least 4 constants |  | Intro/ description **(1)**   * Clarity / Useful info |  |
| Added Features @ 2 **(4)**   * Clear description / Usefulness |  |
| 1. Use conditional statement to display appropriate messages and reports required  **(10)**    * If-statement : new cases +/- msg    * If-statement : choose report type    * Correct table totals & changes    * Correct output / report II format    * General - constants etc |  | Overall Program Design **(2)**   * Structure Chart, >= 3 level * Correct / Suitable / Tidy |  |
| Method of Solution **(2)**   * Flowchart OR pseudocode * Correct use, eg. Symbols, indentation etc |  |
| 1. Use loop to process data for more than 1 day **(14)**    * Appropriate looping    * Correct total days & to/from day nos    * Highest & lowest cases + day nos    * Suitable ending message |  | Constants & Variables **(1)**   * Correct list pasted from program * table for constants   + Name, value, purpose * table for variables   + Name, data type, purpose |  |
| Added Features **(6)**   * 2 extra features @ 3 marks   o Usefulness to user/customer o Complexity of program/logic o Successfully implemented   * NOT considered as features - use of program constructs / commands (eg. Blinking screen) |  | (Screenshots showing the added features must be included somewhere in the chapter, and indicated)  Outputs - Run 1 **(1)**   * description * Scenario table (new page) * Complete set of screenshots |  |
| User Friendliness **(4)**   * Prompts & Responses * General courtesy * Overall Screen Layout is tidy |  | Outputs - Run 2 **(2)**   * description * Scenario table (new page) * Complete set of screenshots |  |
| Readability **(10)**   * suitable identifier names * constants used as appropriate * adequate & useful comments * program indentation * clear programming constructs |  | Outputs - Run 3 **(2)**   * description * Scenario table (new page) * Complete set of screenshots |  |
| Originality/Creativity **(6)**   * uniqueness * creativity |  | Late Penalty   * less 10% per day |  |
| Penalty (if any)  **(@ -2)**   * Inefficient / illogical structures, eg:   + unnecessary looping   + inappropriate structures * -2 per occurrence (at tutor's discretion) * Late submission for A, B, C @ -2 per day |  | Other Penalty (if any) **(@ -1) / ( -5)**   * Missing / incomplete item/chapter   + -1 per item * Flowchart/pseudocode is totally different from program   + -5 (at tutor's discretion) |  |
|  |  |  |  |
| **Total =** | | |  |

**Declaration of Originality**

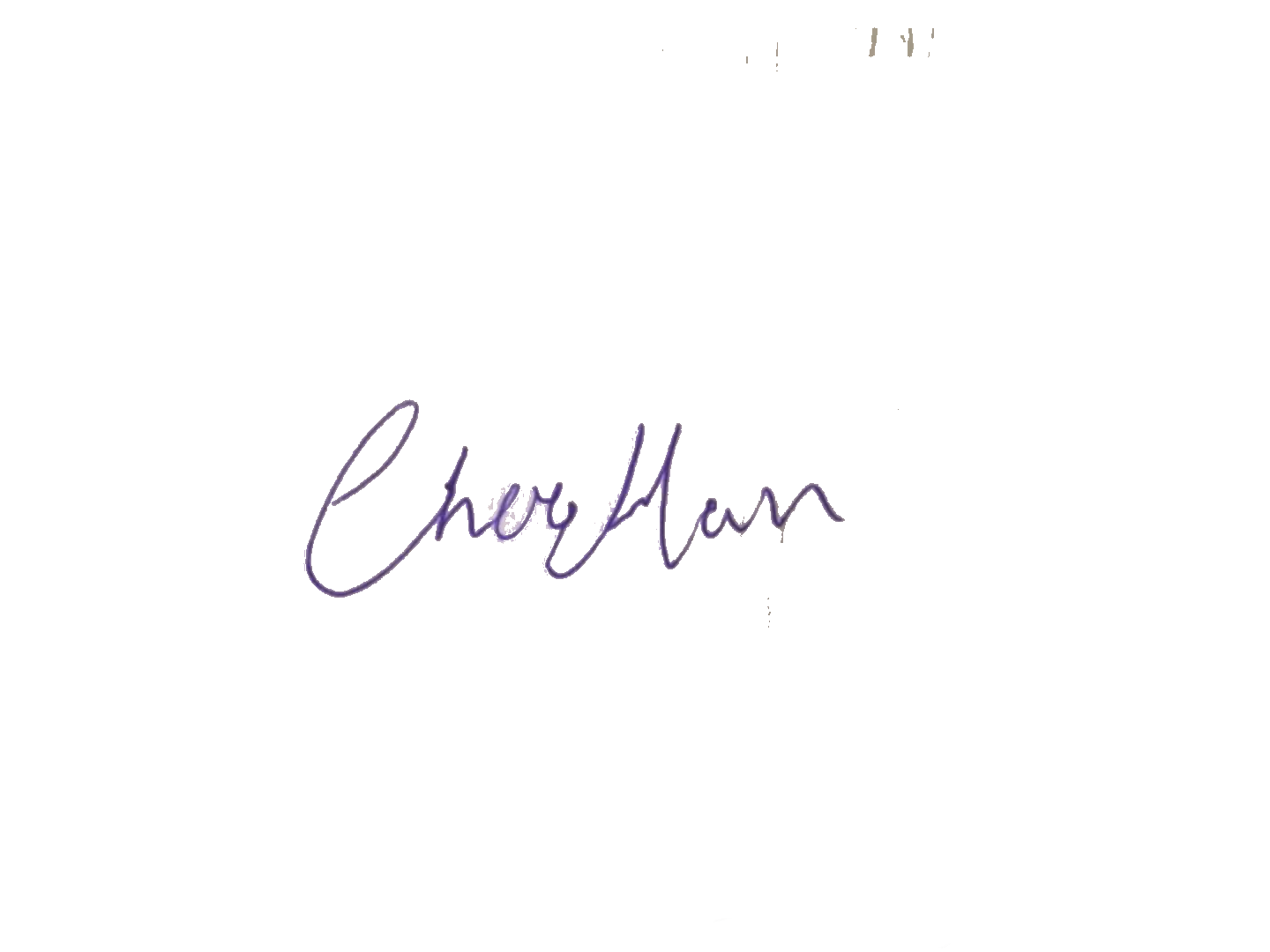
I declare that this assignment is free from all forms of plagiarism and for all intents and purposes is my own work. I understand that I will be penalized if I have not complied with TAR UC’s Plagiarism policy.

****



<Insert your photo>

Student Name : \_\_\_\_POH CHOO HAN\_\_\_\_\_\_\_



Signature :

Date : \_\_\_\_\_29th July 2020\_\_\_\_\_\_\_\_\_

**Table of Content**

|  |  |  |
| --- | --- | --- |
| **No.** | **Title** | **Page No.** |
| 1 | Brief description / purpose | 1 |
| 2 | Overall program structure | 2 |
| 3 | Method of solution | 3 ~ 16 |
| 4 | Added features | 17 ~ 18 |
| 5 | Program testing & outputs | 19 ~ 43 |
| 6 | Constant & variables | 44 ~ 49 |

1. **Brief Description / Purpose**

This program did help the local authority to keep on tracking the data regarding the disease cases throughout this covid-19 pandemic. This is a Covid-19 Cases Tracking System that did allow the local authority to key in the cases data effectively. At the end of inputting the data of a day, the program is able to produce a summary report or comparative report which help the authority to monitor and make sure the pandemic is under control.

The authority will be able to key in the details include the number of local and imported new cases, number of deaths, and number recovered. The data is then processed and will be able to produce a summary report which showing the number of new cases, total cases, total deaths, total recovered, total under treatment and those corresponding rates. Other than that, there are also comparative report which allow the user to compare the current data with yesterday’s data.

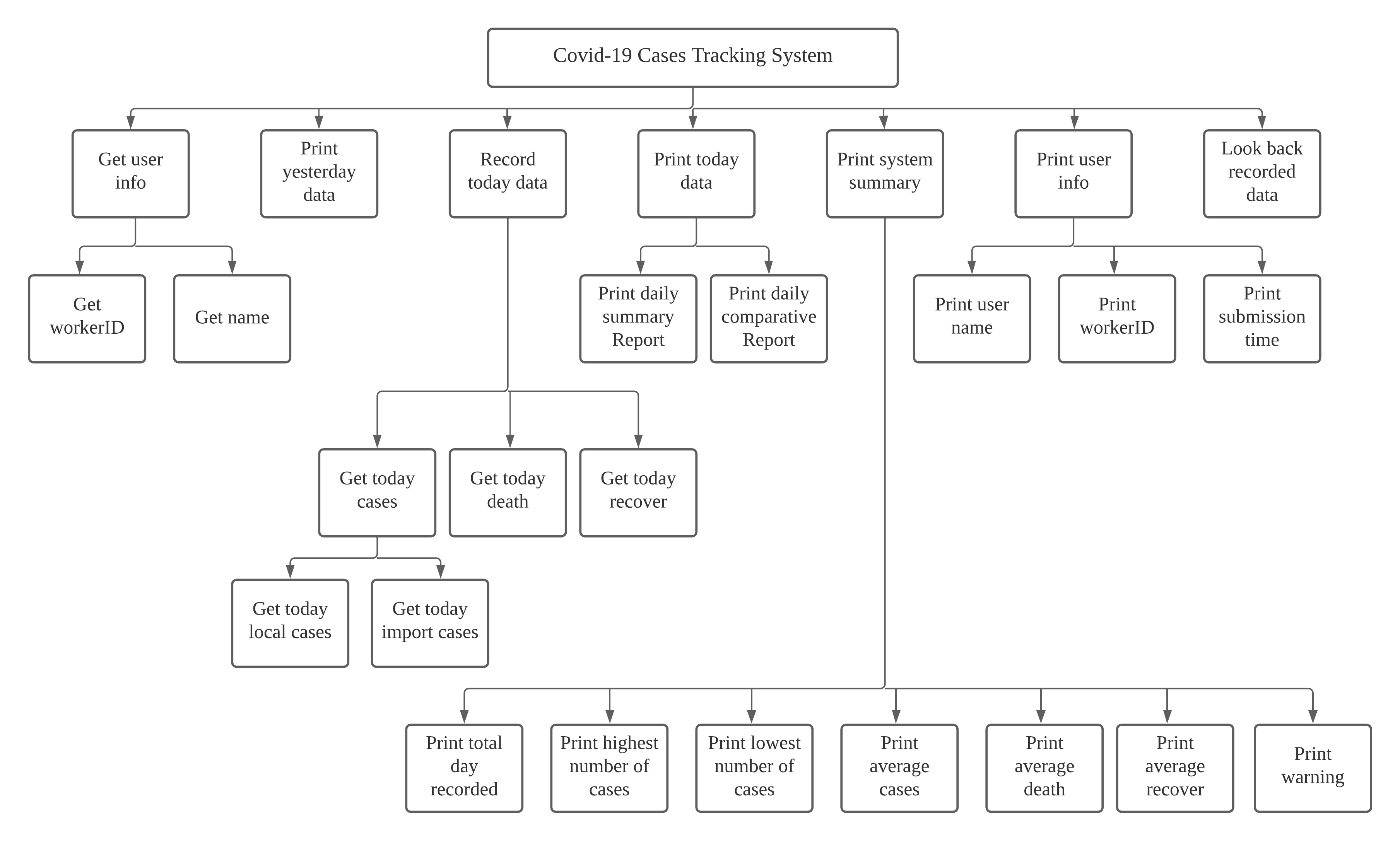
The user is able to carry on key in the further day’s details continuously. After the last day’s report, the program will show how many days are recorded together with the starting and ending days numbers. There are also the highest and lowest number of daily cases and their respective day on which it occurred.

At the end of the report, it will print the total days recorded, highest number of cases, and lowest number of cases. Other than that, the program itself will calculate the average cases, deaths, and recover. If the average cases are more than 100 it will show yellow warning and more than 500 will print orange warning. Last but not least, it will show the submit time and user’s details for the authority to keep on tracing the input.

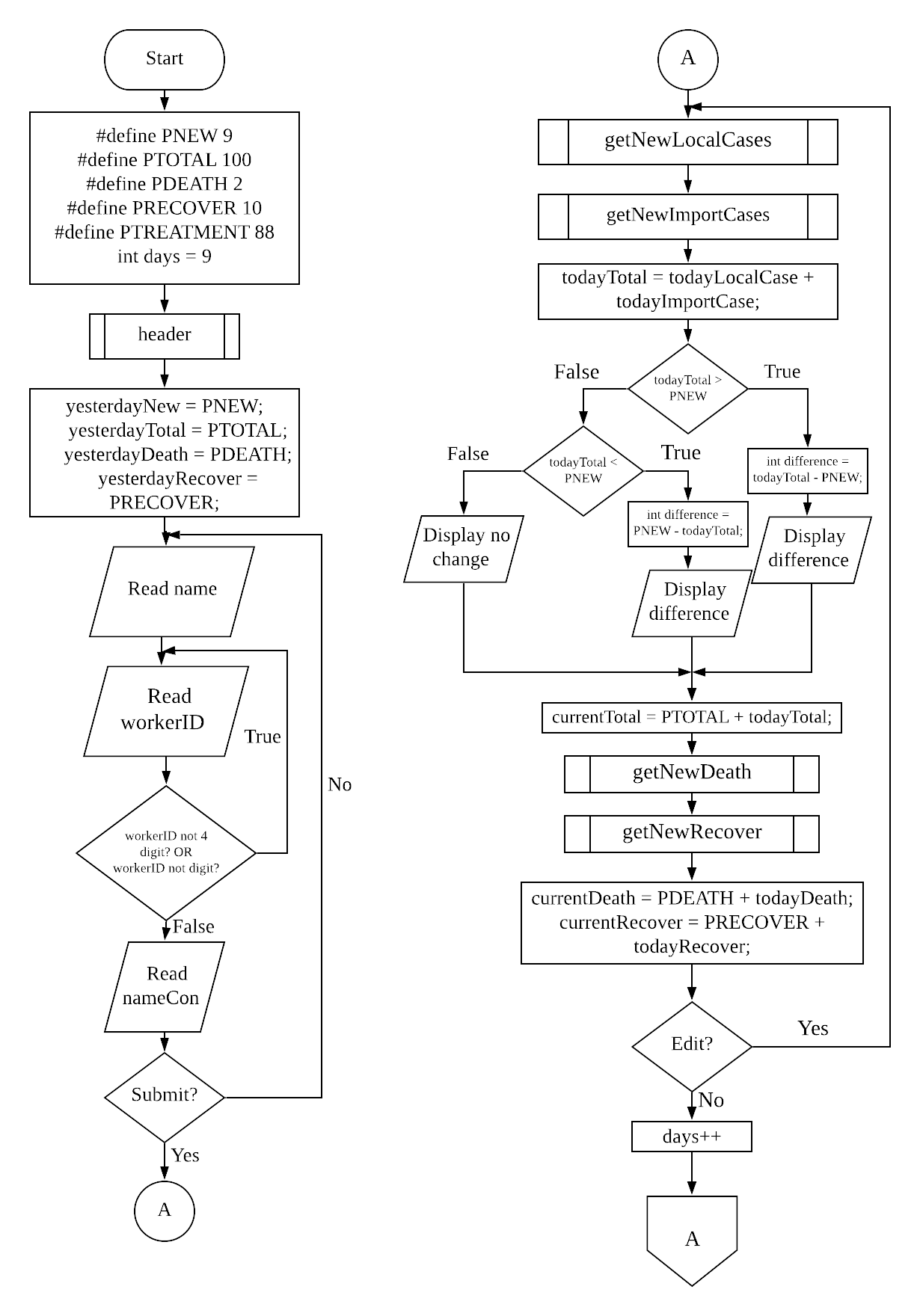
When getting the input from the user, the system will make sure that every single input that user gave is accurate and valid, otherwise it will be been rejected. Another response is asked in this case. User is require to enter his/her personal identity before he/she start the program, and submission time will together being print at the end of the program. If user find that he/she make a mistake when inputting data, he/she is allow to edit the data. At the end of the program user will be allow to review back the data key-in before.

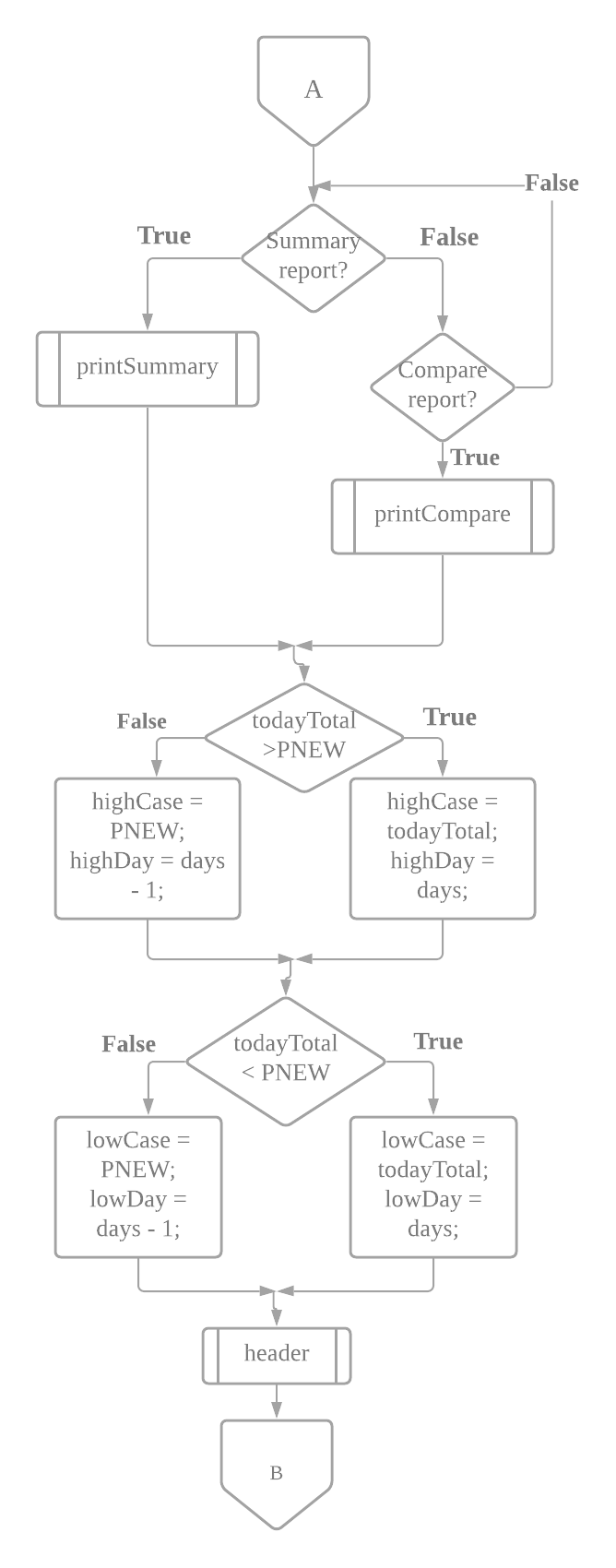
The program allows the local authority to be able to analyse the current status of pandemic. They can get a variety of information from this program, for example the average, the rates, and etc. The local authority could plan their action based on the information provided so that the action is accurate, efficient and effective.

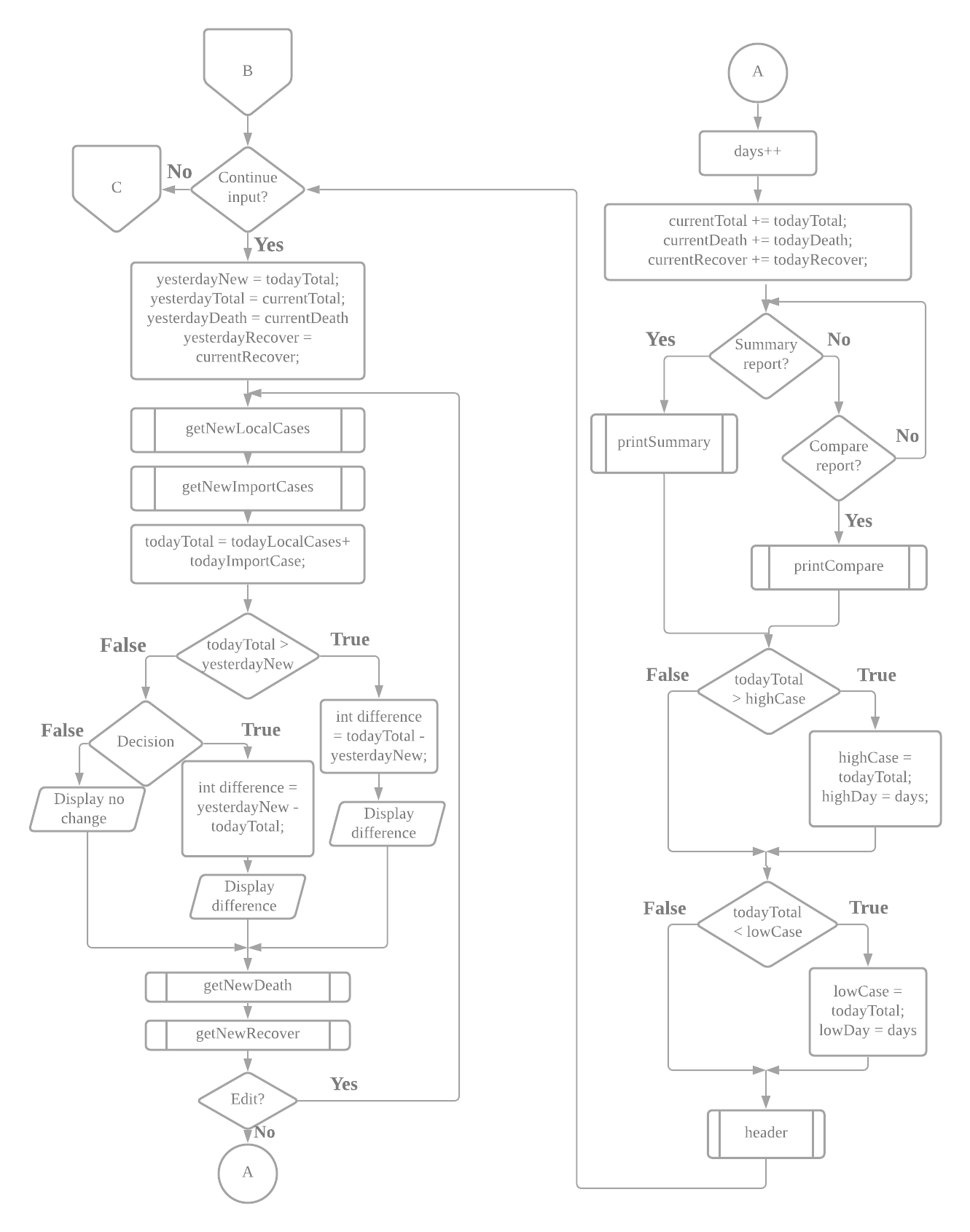
1. **Overall Program Design**

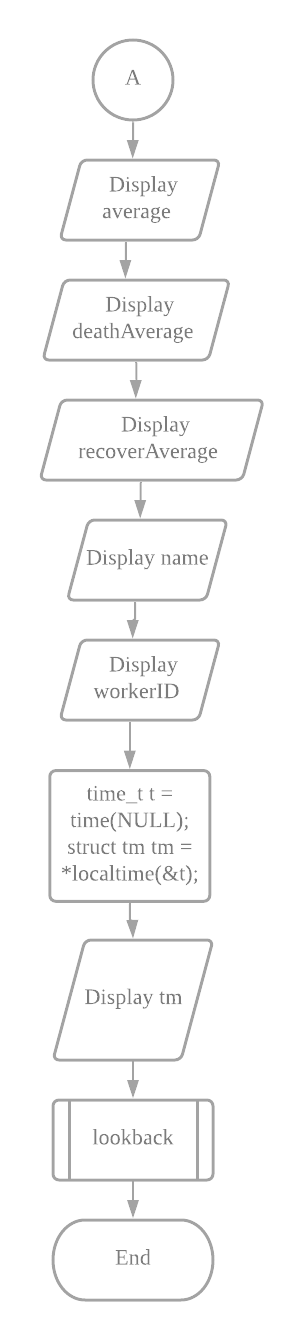
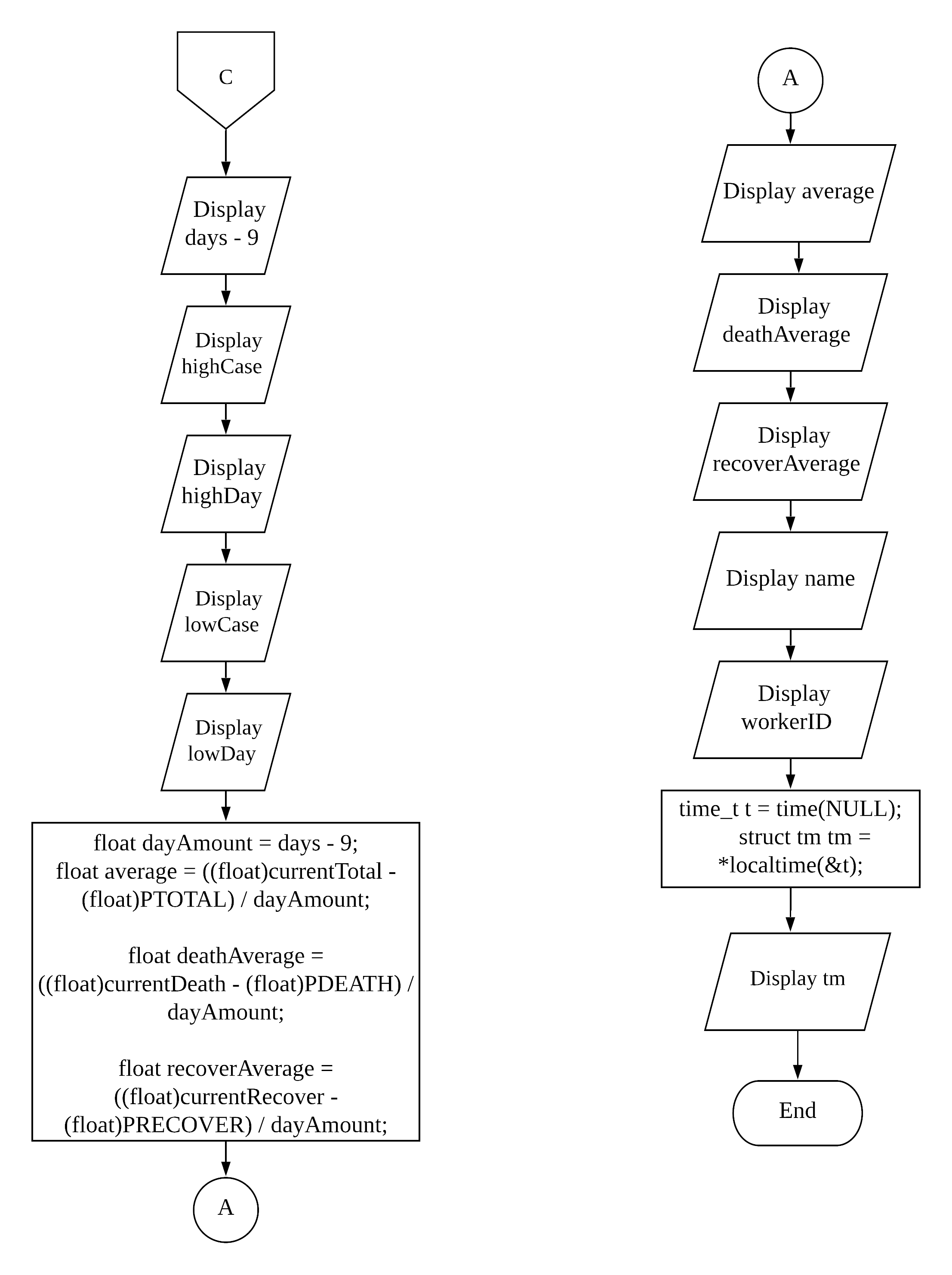


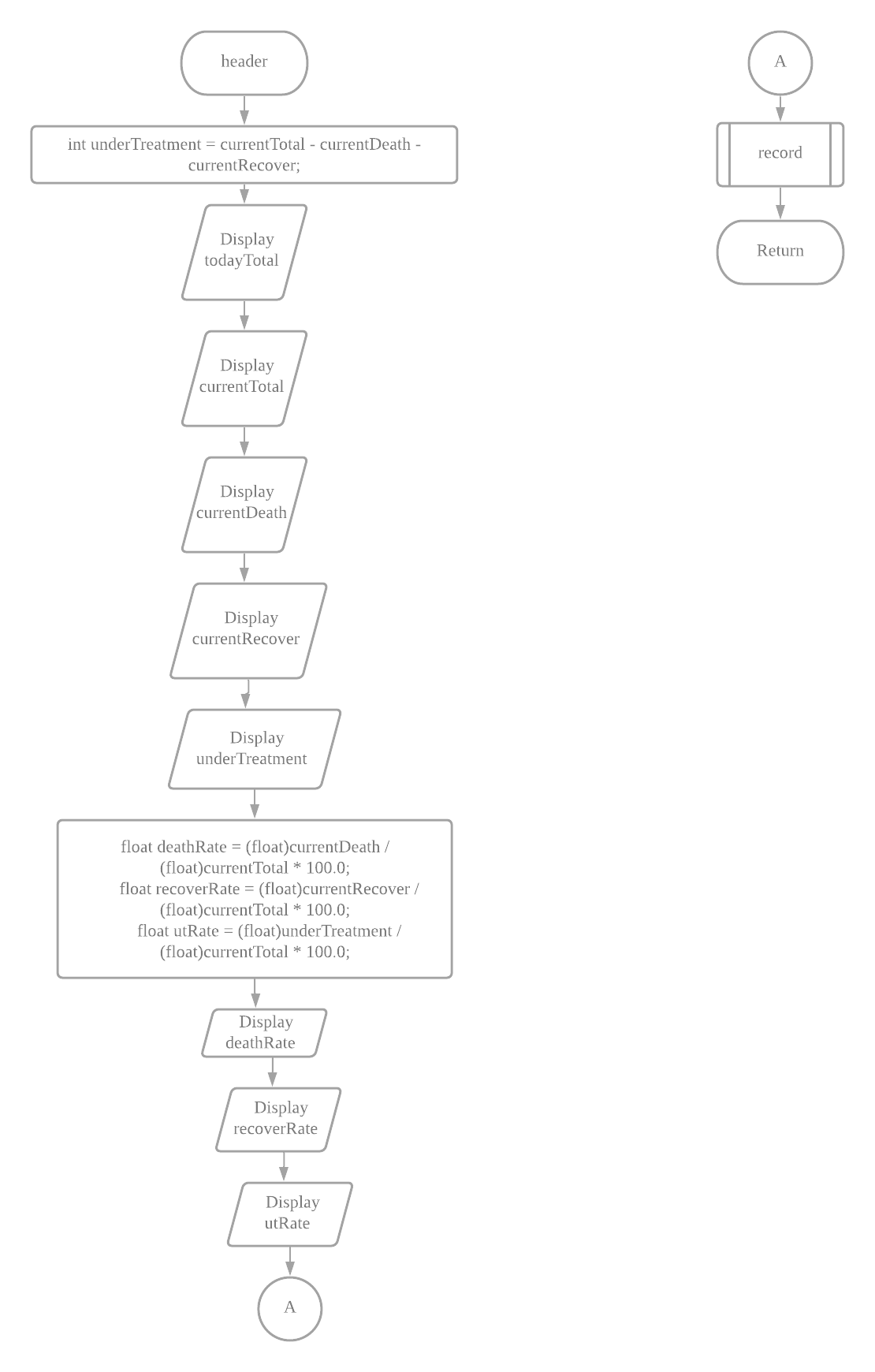
1. **Method of Solution**

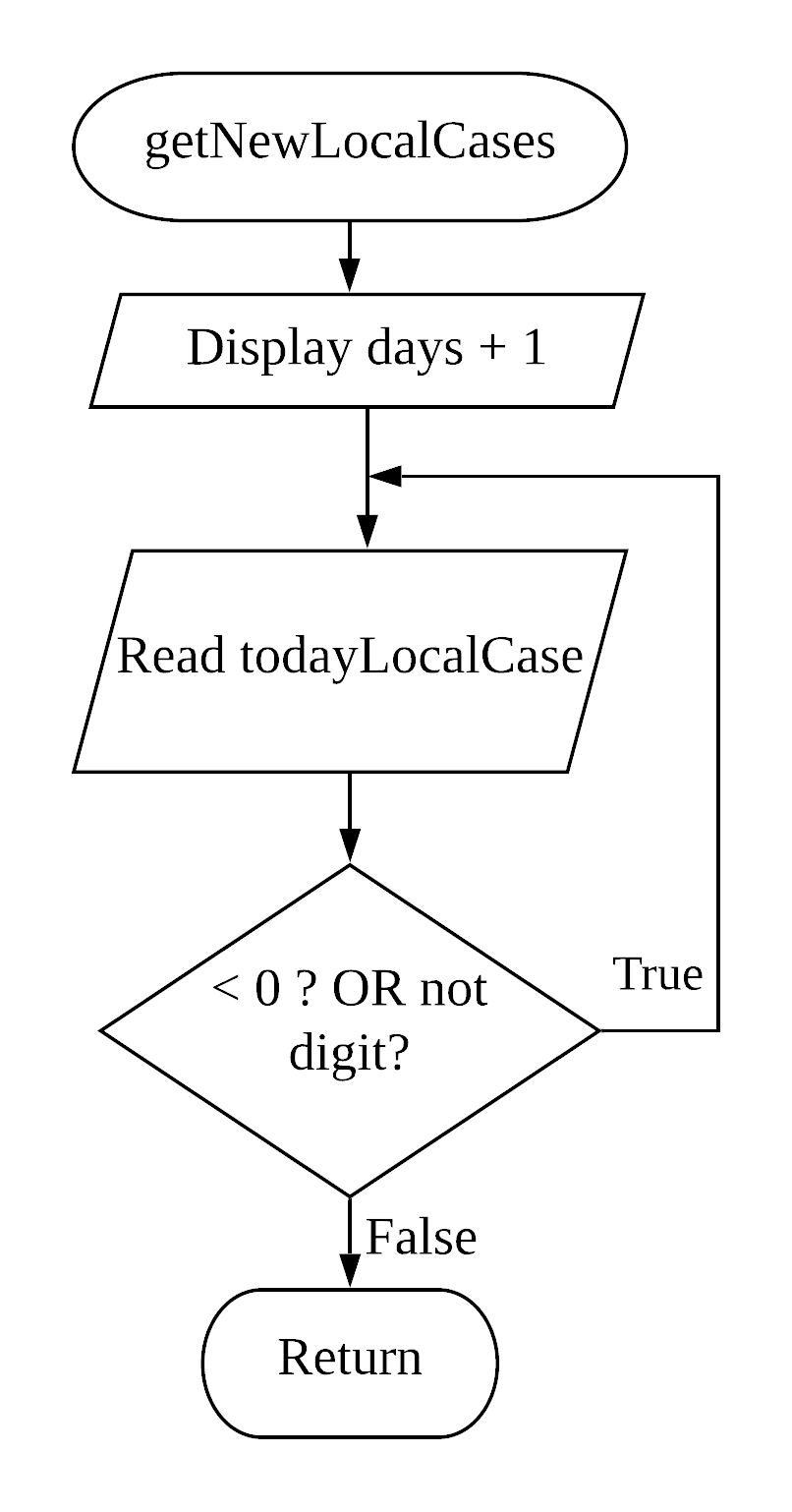


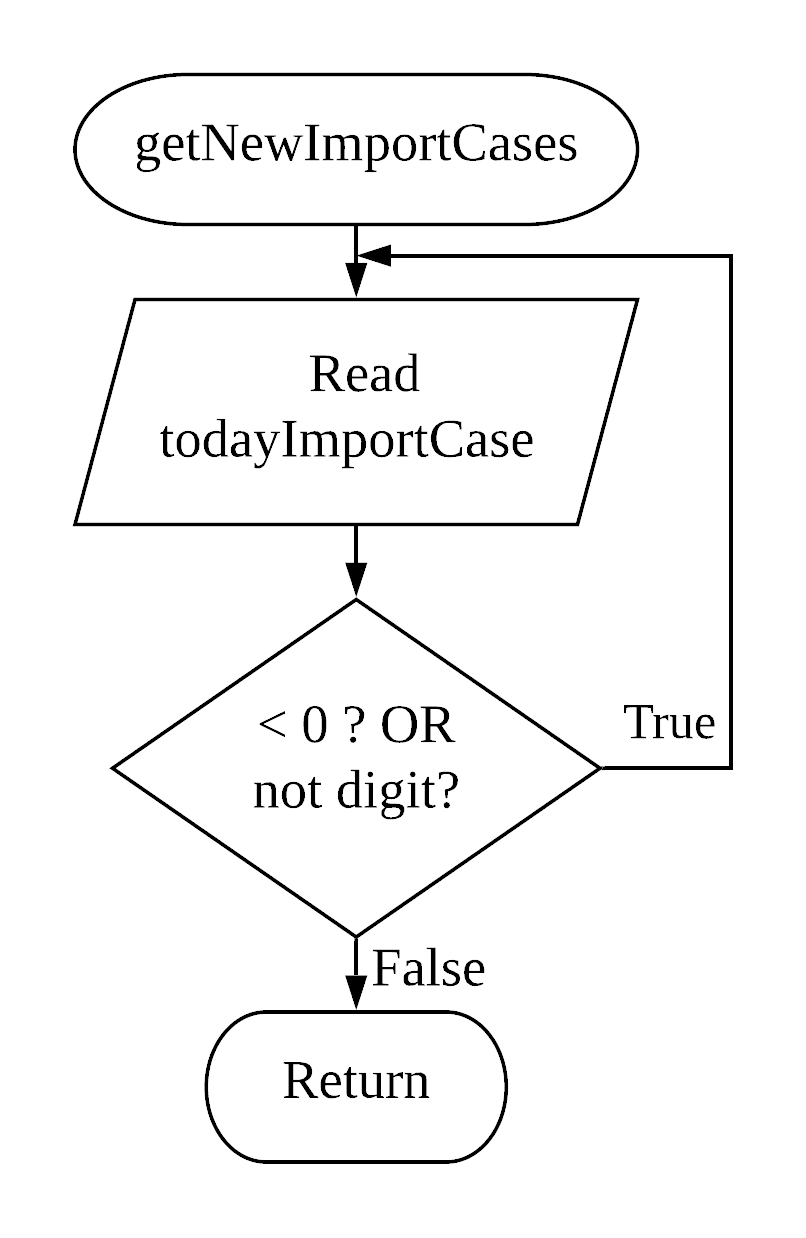


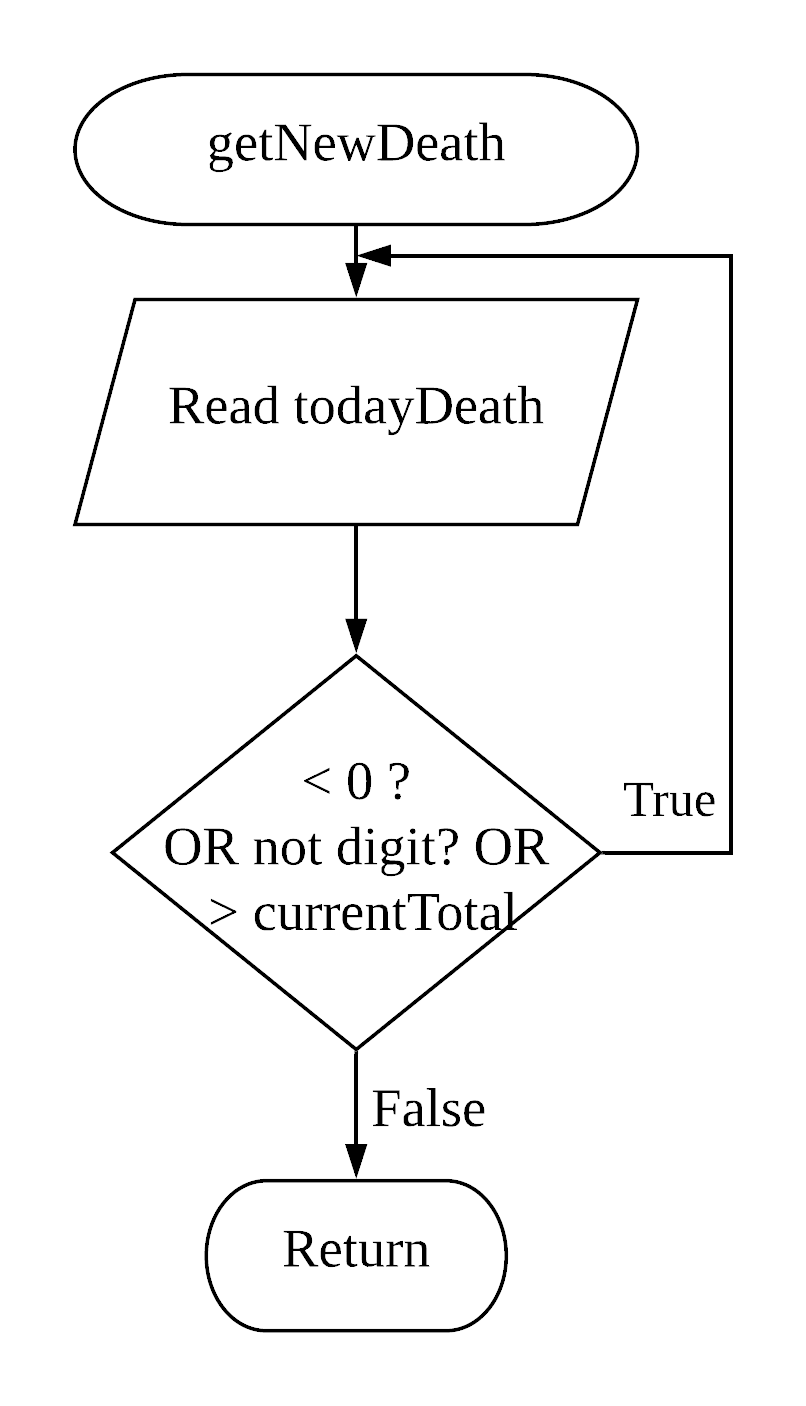


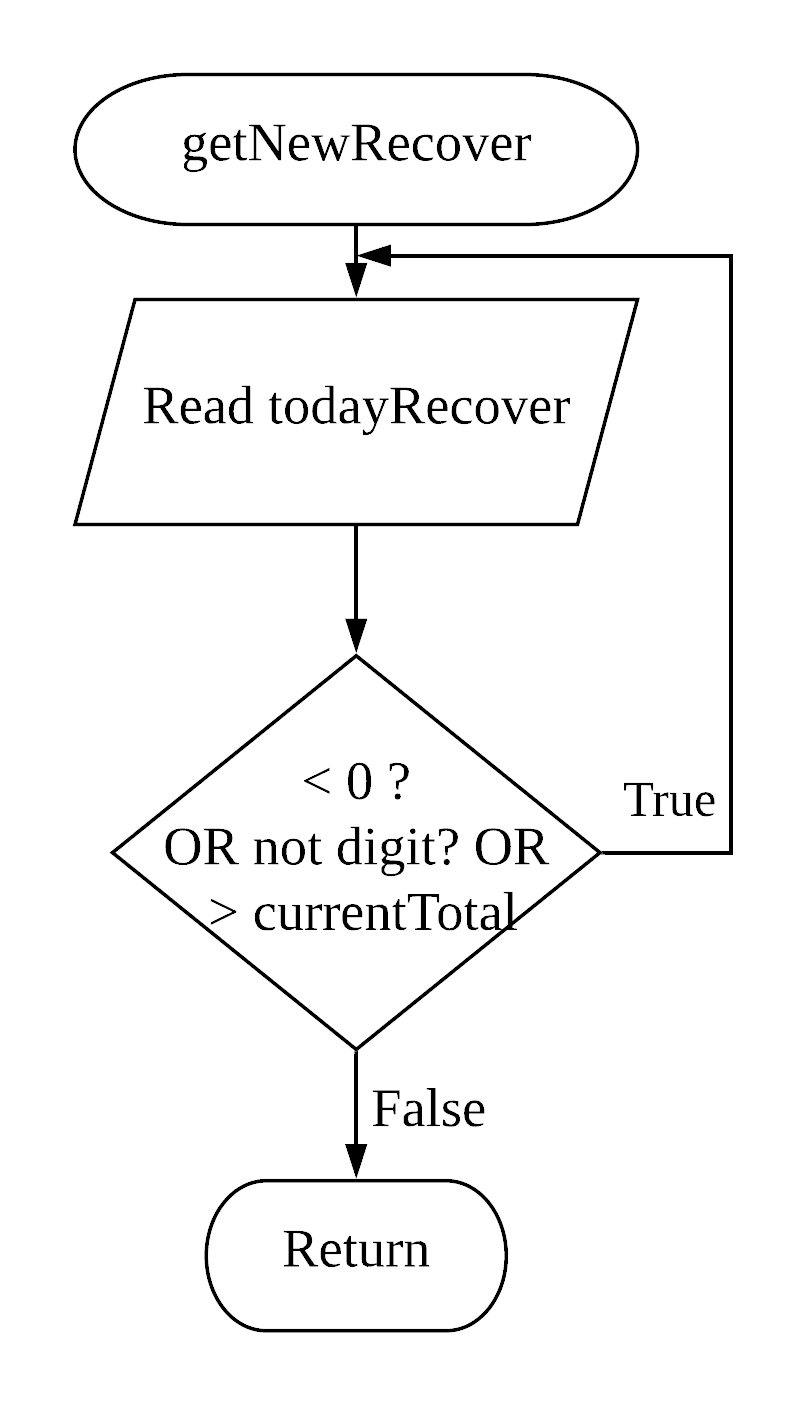


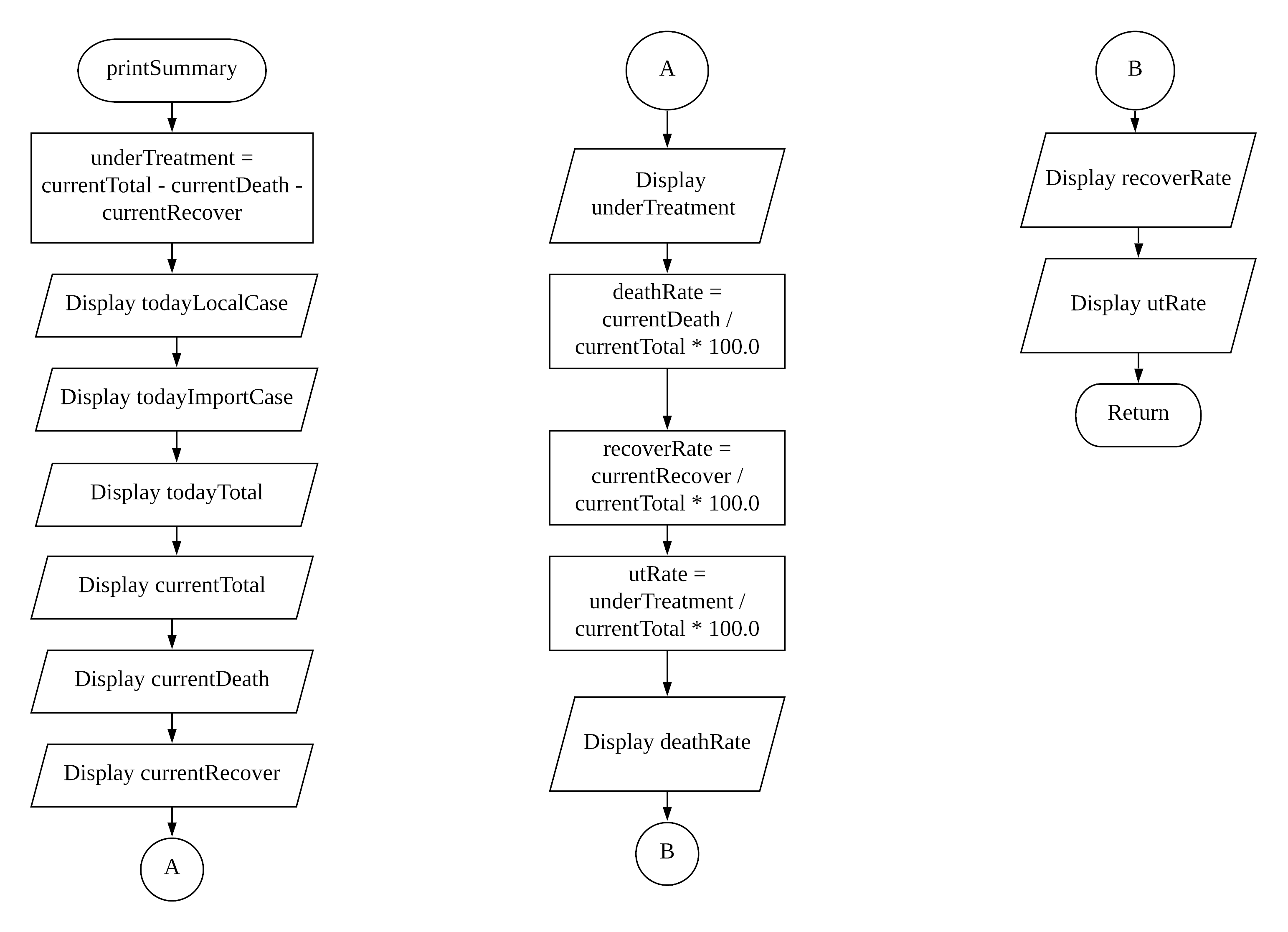
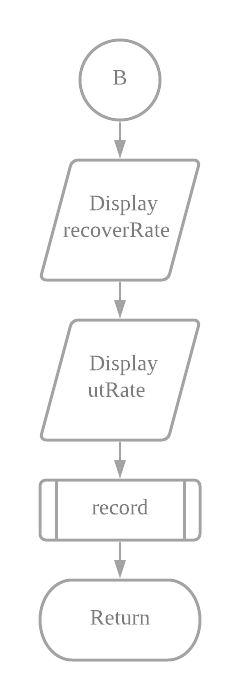


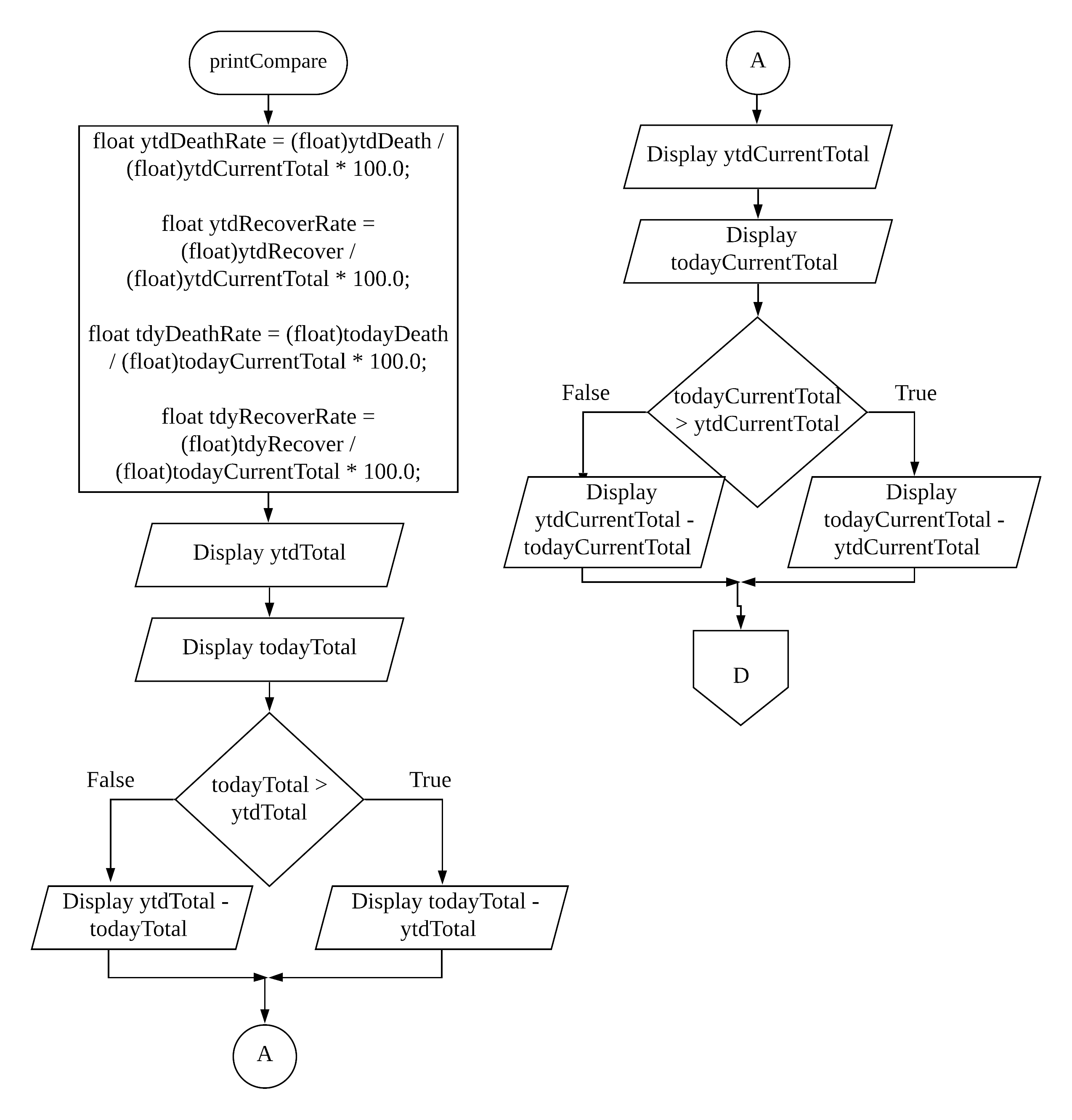


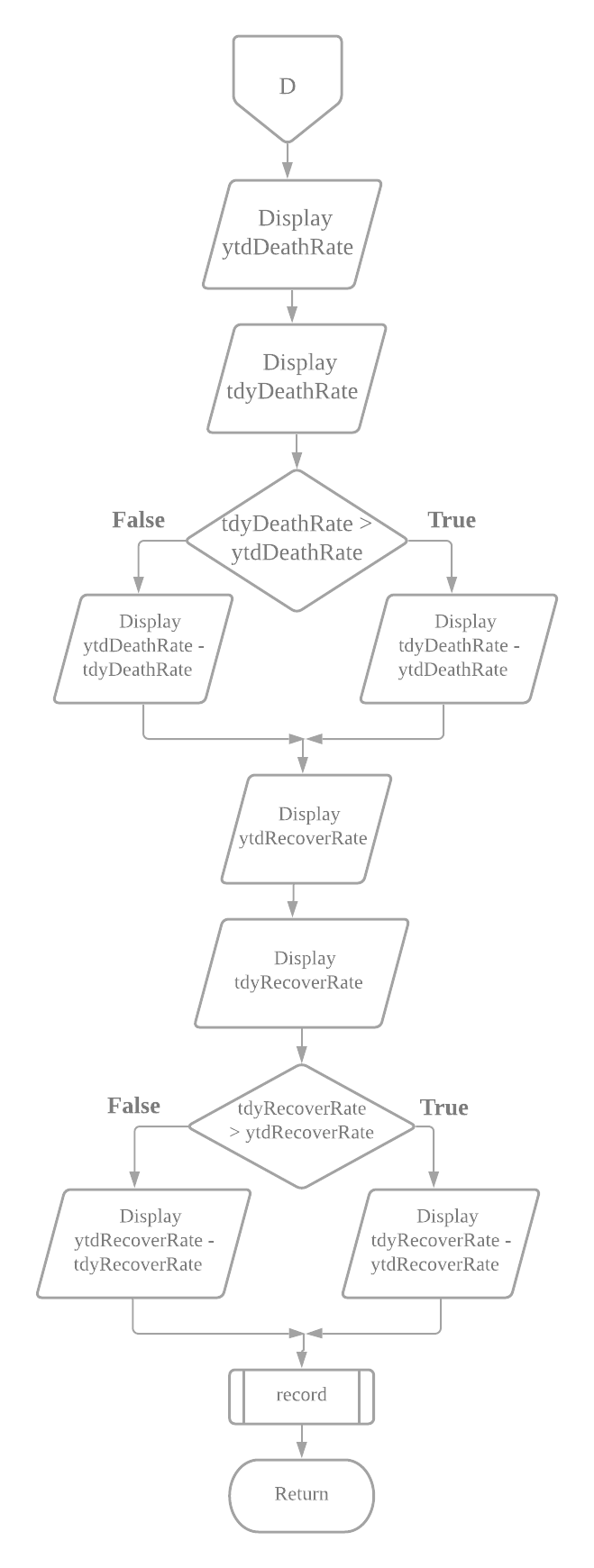


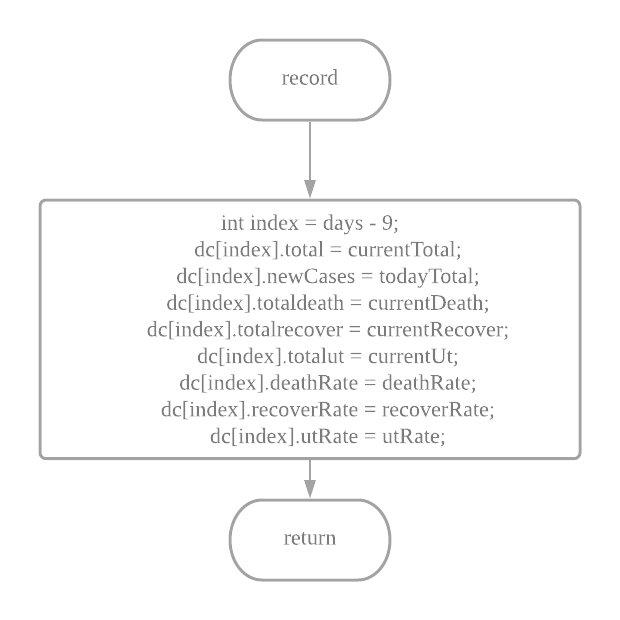
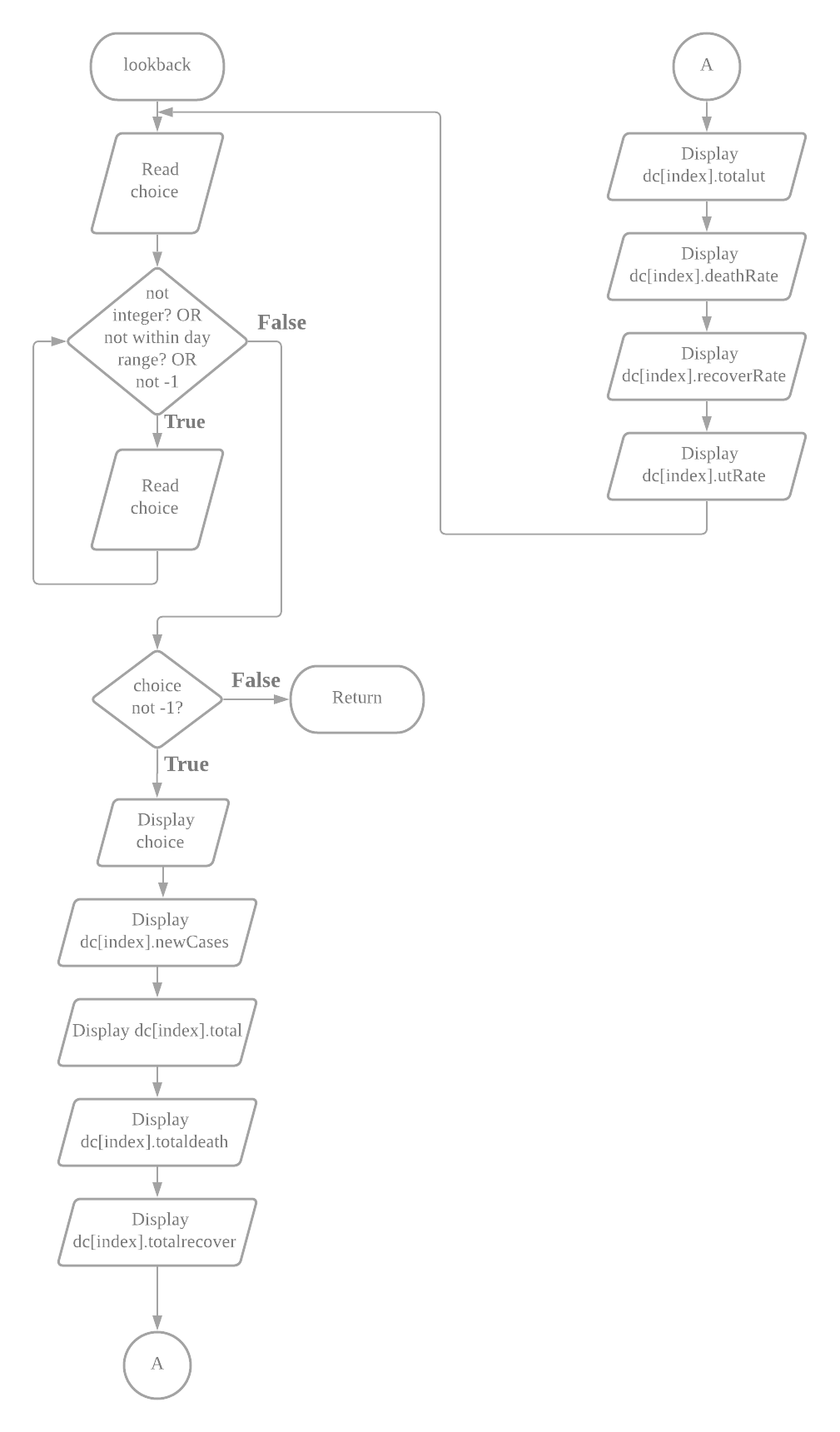












1. **Added Features**

**4.1 User’s name, user’s id and submit time**

Before the user is able to key in the data, the user needs to input their full name and worker ids. After the program come to the end, it will print the worker’s name, id and time of submissions.

This allow the authority to know who is the one key in the data and who is responsible on it. If there are any errors the authority knows who they are going to consult. The submission time tells the authority when the worker submits so that it is good to save as history for future uses.  
  
time.h is used to get the submission time and the time will be shown. User name and id is recorded at the beginning of the program and it will print out after the end of day recording.

* 1. **Average cases, death, recover and warning message for overall inputted days**

After all the input process is end, it will show the user about the average cases, death, recover and warning message. The warning message is send depending the average with the following scenario:

|  |  |
| --- | --- |
| **Average cases** | **Messages** |
| More than 500 | Orange warning (xxx), government must carry out necessary actions! |
| More than 100 | Yellow warning (xx), government should carry out some actions! |
| Less than or equal to 100 | Green warning (x), it is under control but still require monitor! |

With these data and messages, it did alert the authority when the pandemic is loss of control. The authority can take further action depending on these analysed data and messages!

If-else is use to determine which warning, and show the warning respectively. Average cases is obtained by dividing total cases between day recorded and the total day recorded.

* 1. **Editable data after user input the data.**

After finish entering the data for the specific day, user is prompt to check whether he/she want to rewrite the data or not. If yes, user is allowed to rewrite the data for the day!

It is good that everyone might make a mistake during performing task, so user can edit the data that they inputted if they think they did key in the wrong one or made a mistake. It did improve the accuracy of this program and reduce the mistake!

System will allow user to reinput the data if user response yes on edit. Only the last data inputted for the day will be recorded.

**4.4 Validation of the input**

To avoid some mistake, this program did check the validity for every input the user did. The following scenario is considered:

|  |  |
| --- | --- |
| **Scenario** | **Valid Input** |
| User’s worker id | Should be 4 digits (larger than 999 and smaller than 10000) |
| Edit option | Y, y, N, n. (Other input is not accepted) |
| Report type | 1, 2 (Other input is not accepted) |
| New local case / New Import case / No. of deaths / No. recovered | Any positive numbers and 0 (Negative numbers are rejected) |
| No. of deaths / No. recovered | Cannot more than overall total cases |
| No. of deaths / No. recovered / New local case / New import case / Workerid or any integer type input | Reject any value except integer type. |
| All kind of input except name | Space between words is rejected, it only allow one value can be input. |

It did help prevent the user to key in an invalid data that may reduce the accuracy or corrupt the program. If the inputted data is invalid, the system will prompt again by using loop until the data is valid!  
  
User input is check and reject if the input is not acceptable. User will be able to re-enter the value without limit.

* 1. **Look back the recorded day’s cases**

After the user stop inputting the data, it will bring user into a loop that allow him to check any day’s data that he/she had input. It will continue asking user which day to check until a sentinel value -1 is inputted.

This allow user to review what he inputted at the end of the program so that he can make sure he did not make any errors. It is good to let he/she to double check so that the data is accurate enough. He/she can analyse the result even though the program is nearing the end.

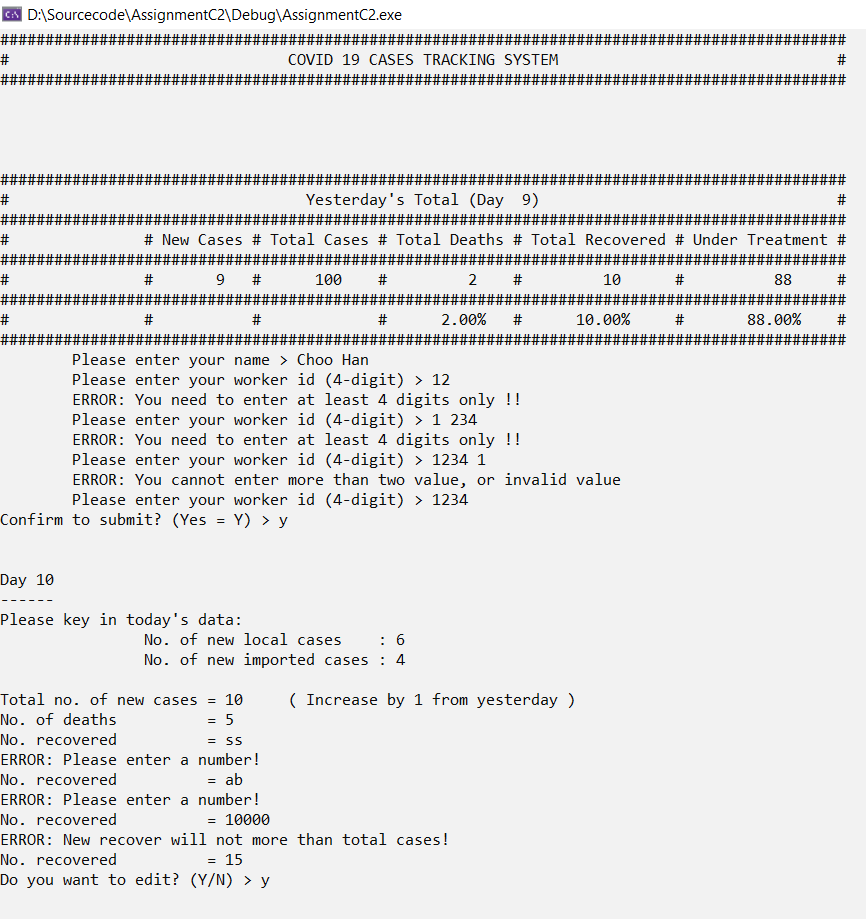
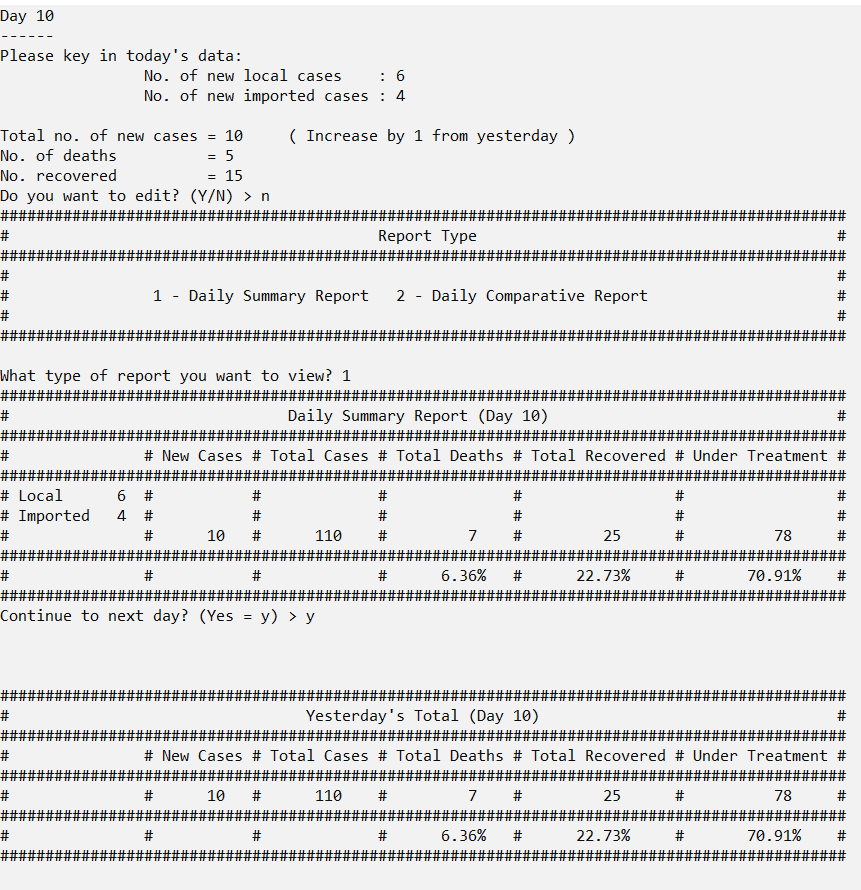
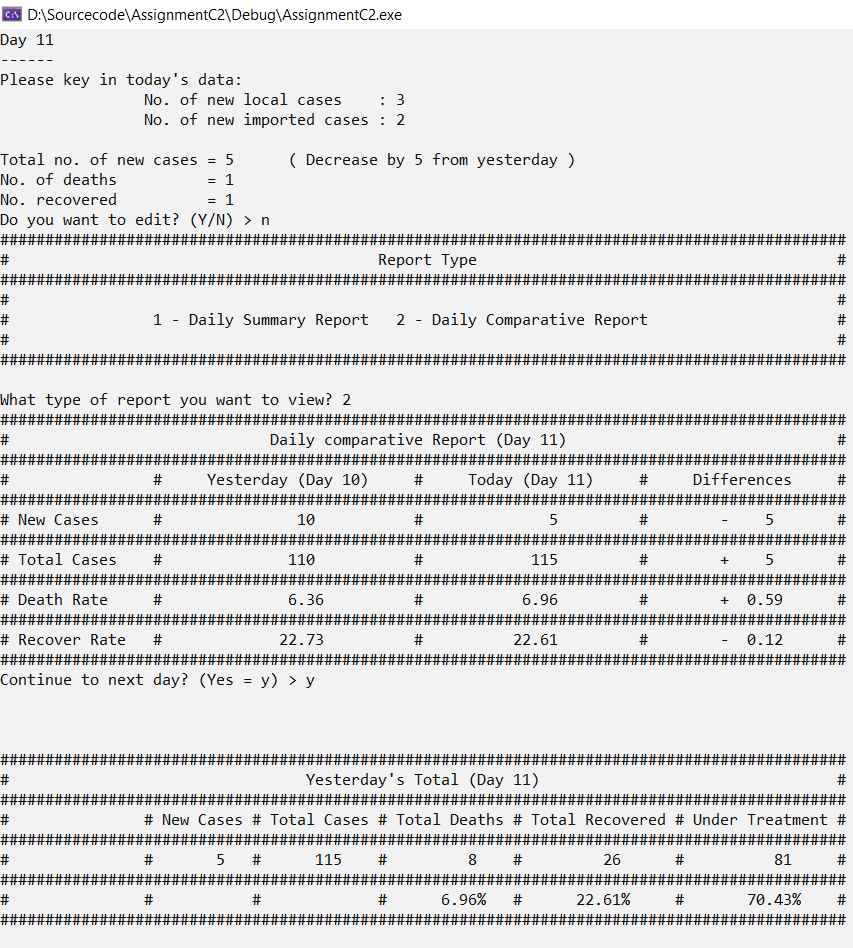
The data is store into the structure array and is been extract when user request for the specified days’ data. User is allowed to extract until he/she wish to stop by entering -1.

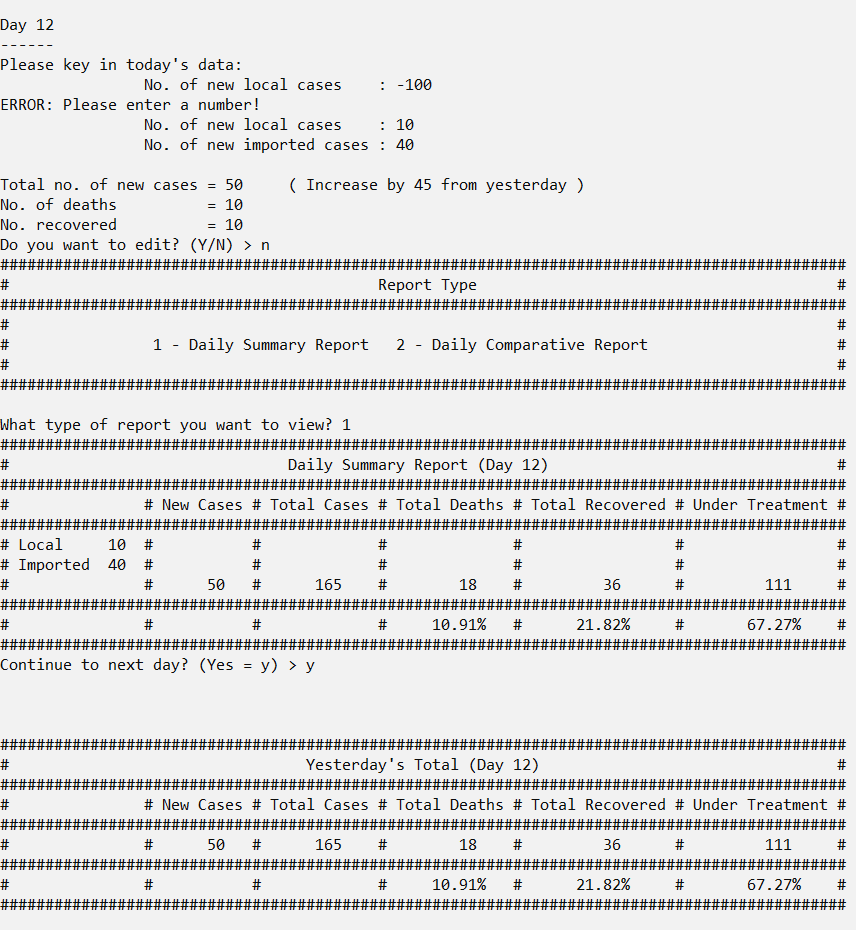
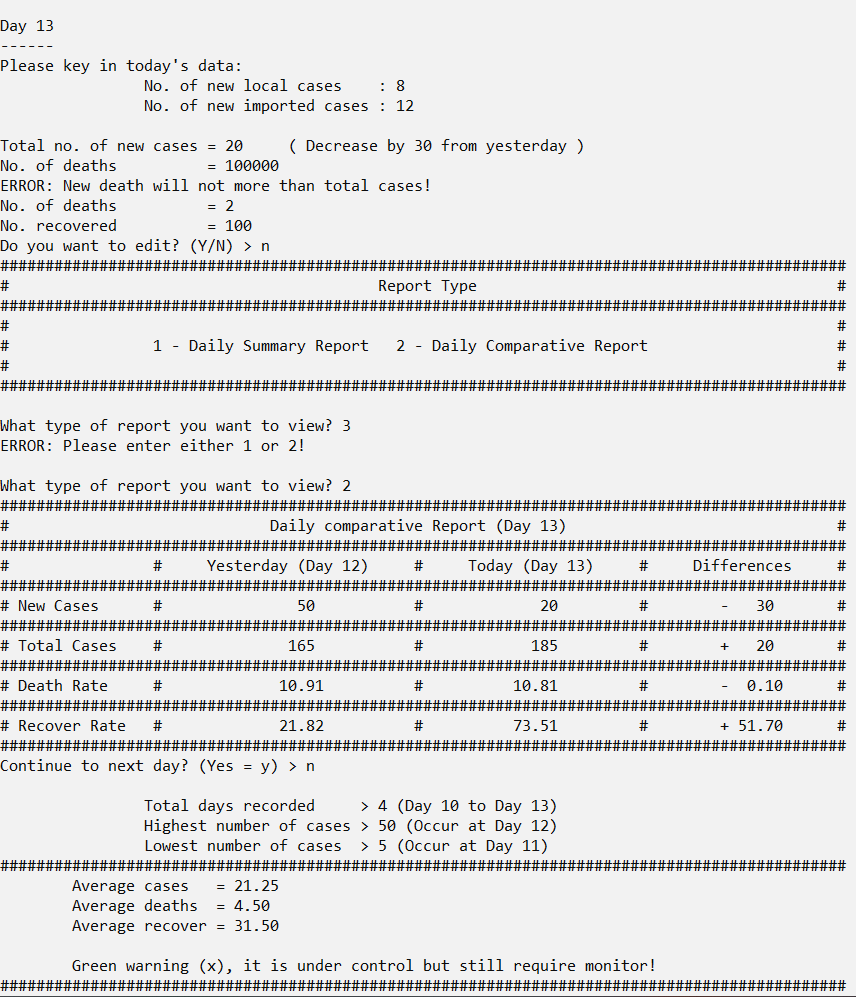
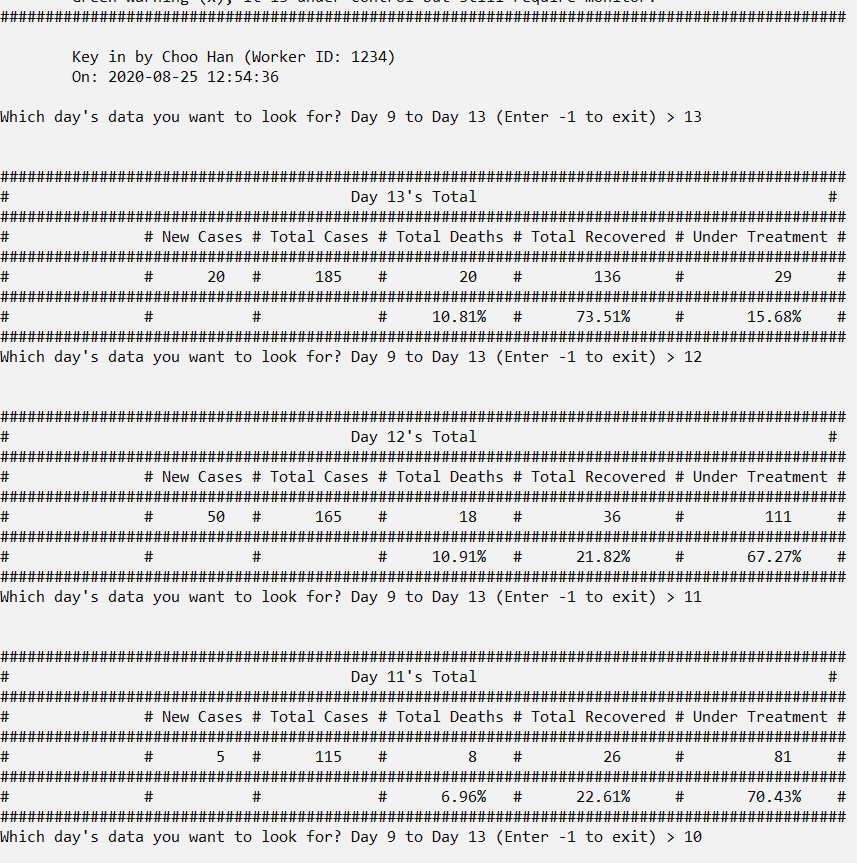
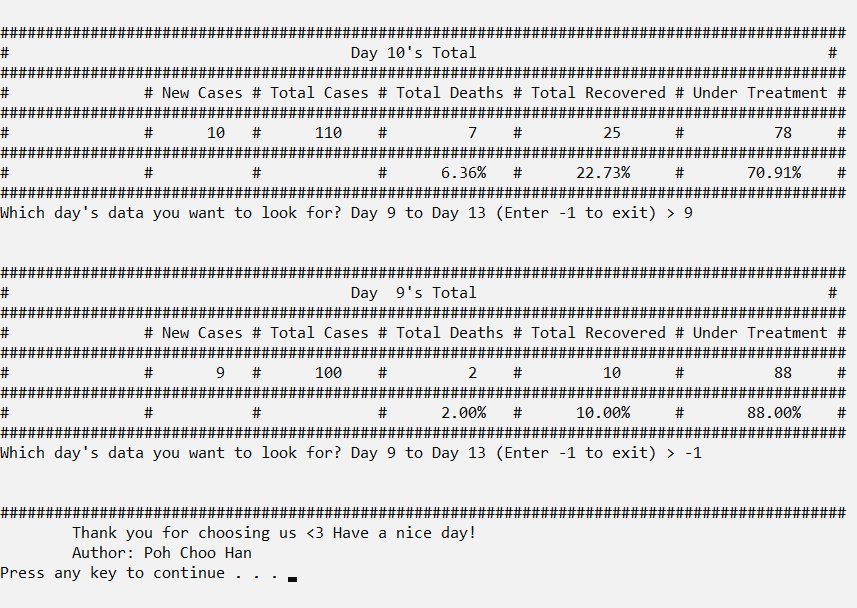
1. **Program Testing & Outputs** 
   1. **Run 1 Scenario**

The data inputted is relatively low to show the basic function of the system. A bunch of error input is try during this scenario. It is much easy for user to input the data, as the system will scan for error input and allow user to retype. The scenario below had test for inputting several invalid inputs. There are total 4 days data is recorded, the highest number of cases is occurred at Day 12 which has 50 new cases, while the lowest number of cases is occur at Day 11 which has only 5 new cases. The grow rate is low so the system will show green warning, which mean Malaysia is relatively safe in Covid-19 pandemic. The added feature (1) user ID, name and time, (2) Average case, death, recover and warning, (3) Editable data (4) Validation, (5) Look back data are demonstrate and test here.

**Test Data + Expected Outputs**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Inputs | | | | Expected Results / Outputs | | | | |
| Loop No. | Day No. | New Local Cases | New Imported Cases | No. of deaths | No. recovered | New cases | Total Cases | Total Deaths | Total Recovered | Total Under Treatment |
|  | Previous  9 |  |  |  |  | 9 | 100 | 2  (2.00%) | 10 (10.00%) | 88  (88.00%) |
| 1 | 10 | 6 | 4 | 5 | 15 | 10  Increase  = 1 | 100 + 10 = 110 | 2+5=7  (6.36%)  Change = 4.36% | 10+15 = 25  (22.73%)  Change = 12.73% | 110 – 7 – 25  = 78  (70.91%) |
| 2 | 11 | 3 | 2 | 1 | 1 | 5  Decrease  = 5 | 110 + 5 = 115 | 7+1 = 8  (6.96%)  Change = 0.60% | 25+1 = 26  (22.61%)  Change =  -0.12% | 115 – 8 – 26 = 81  (70.43%) |
| 3 | 12 | 10 | 40 | 10 | 10 | 50  Increase  = 45 | 115 + 50 = 165 | 8+10 = 18  (10.91%)  Change = 3.95% | 26+10 = 36  (21.82%)  Change =  -0.79% | 165 – 18 – 36 = 111  (67.27%) |
| 4 | 13 | 8 | 12 | 2 | 100 | 20  Decrease  = 30 | 165 + 20 = 185 | 18 + 2 = 20  (10.81%)  Change = -0.10% | 36 + 100 = 136  (73.51%)  Change =  51.7% | 185 – 20 -136 = 29  (15.68%) |
|  |  |  |  |  |  | Ending message:  Total days recorded > 4 (Day 10 to Day 13)  Highest number of cases > 50 (Occur at Day 12)  Lowest number of cases > 5 (Occur at Day 11)  Average cases = 21.25  Average deaths = 4.50  Average recover = 31.50  Green warning (x), it is under control but still require monitor. | | | | |

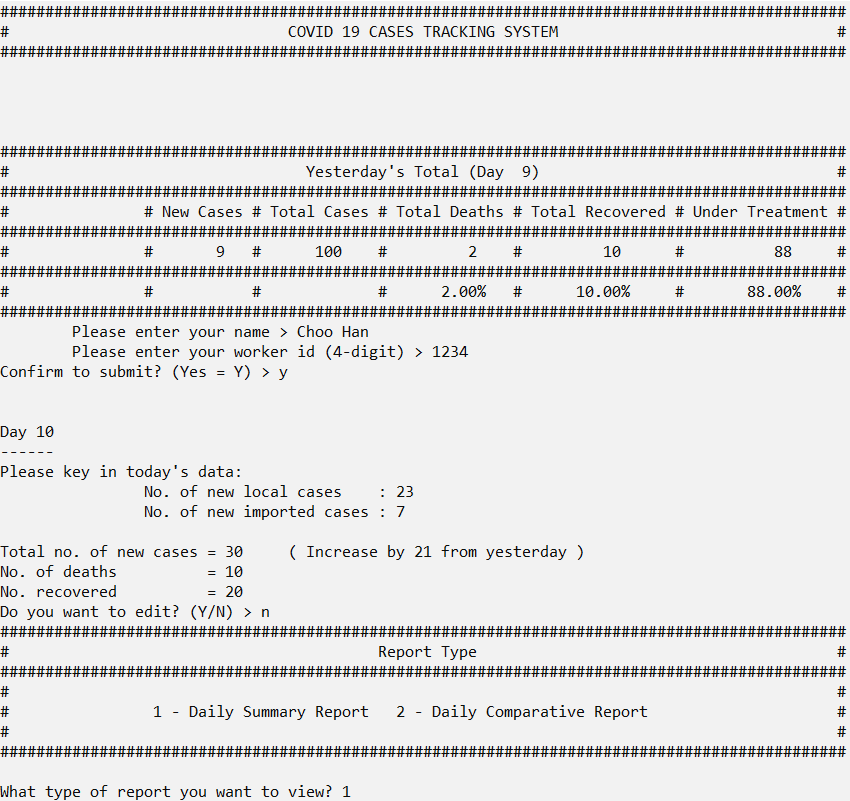
   

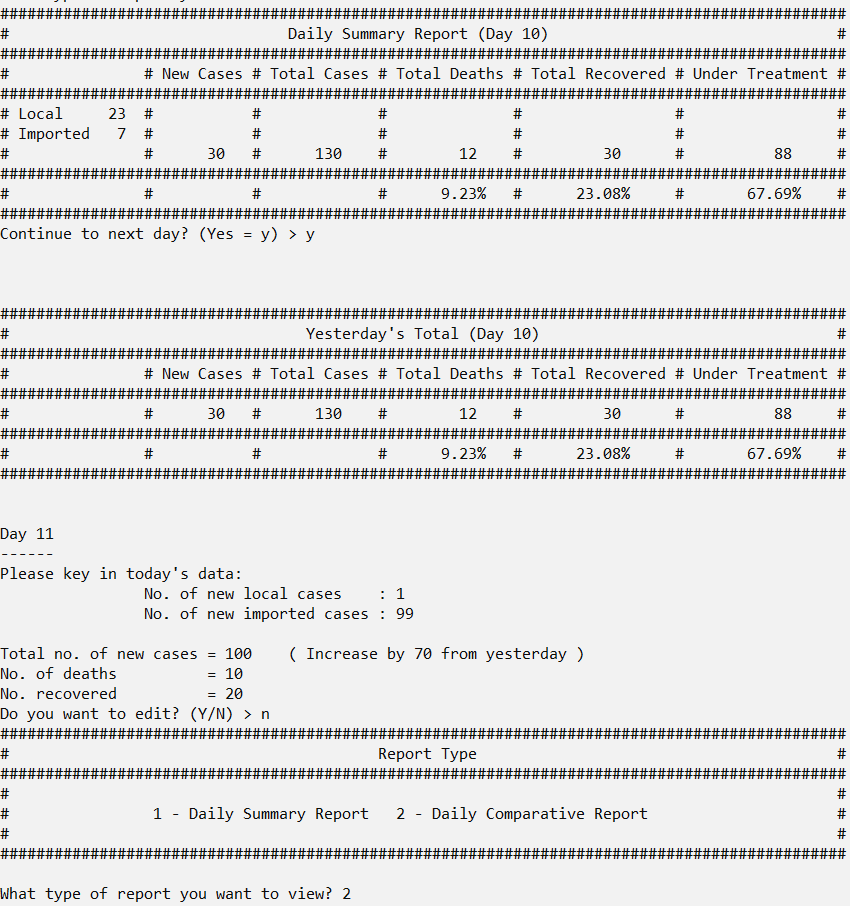
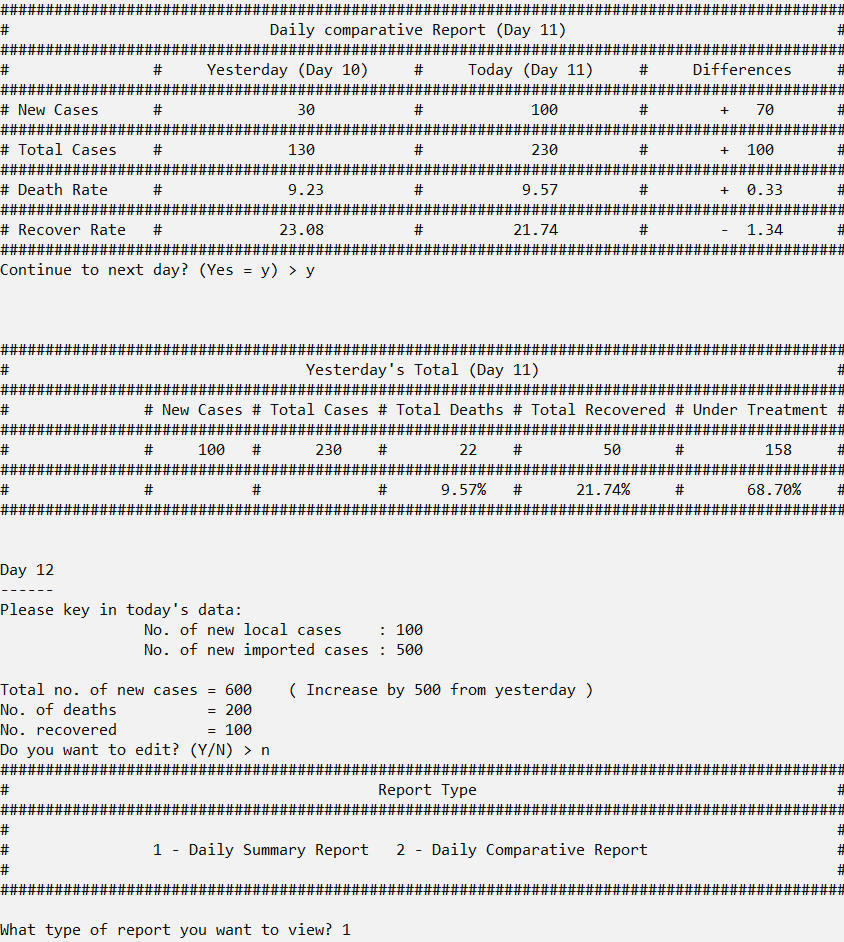
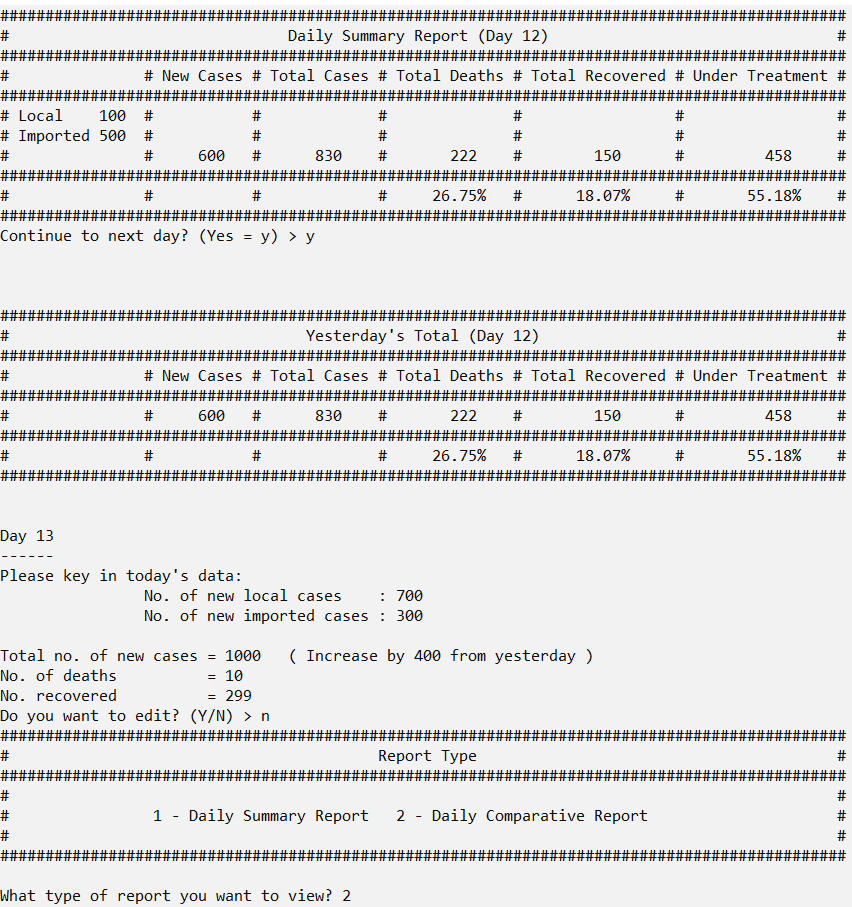
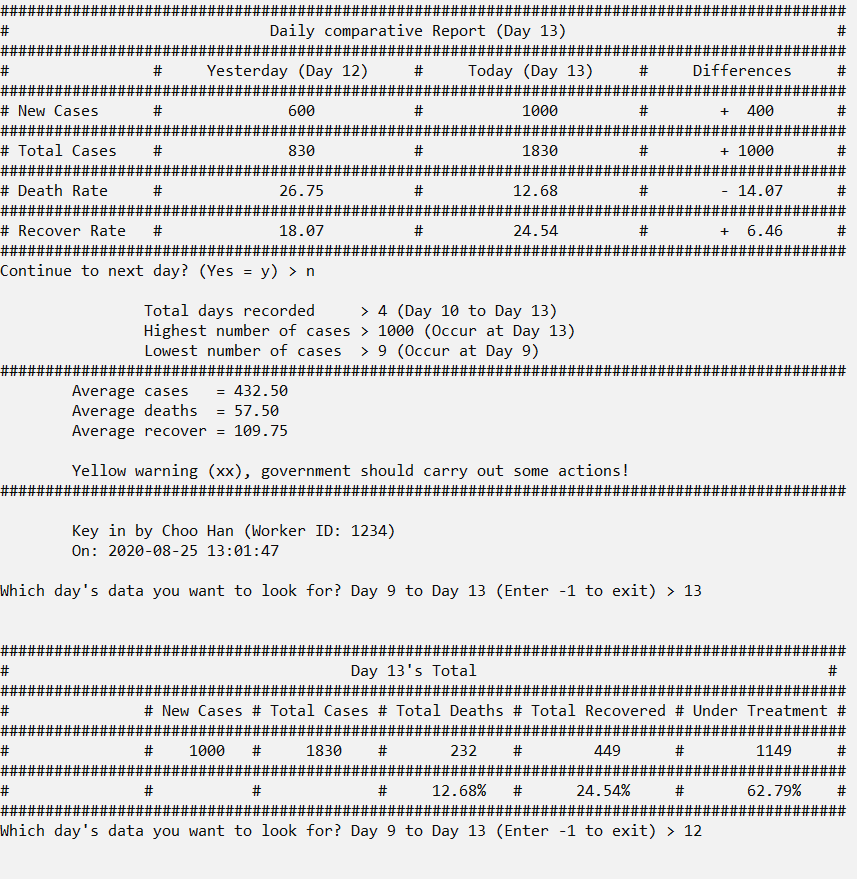
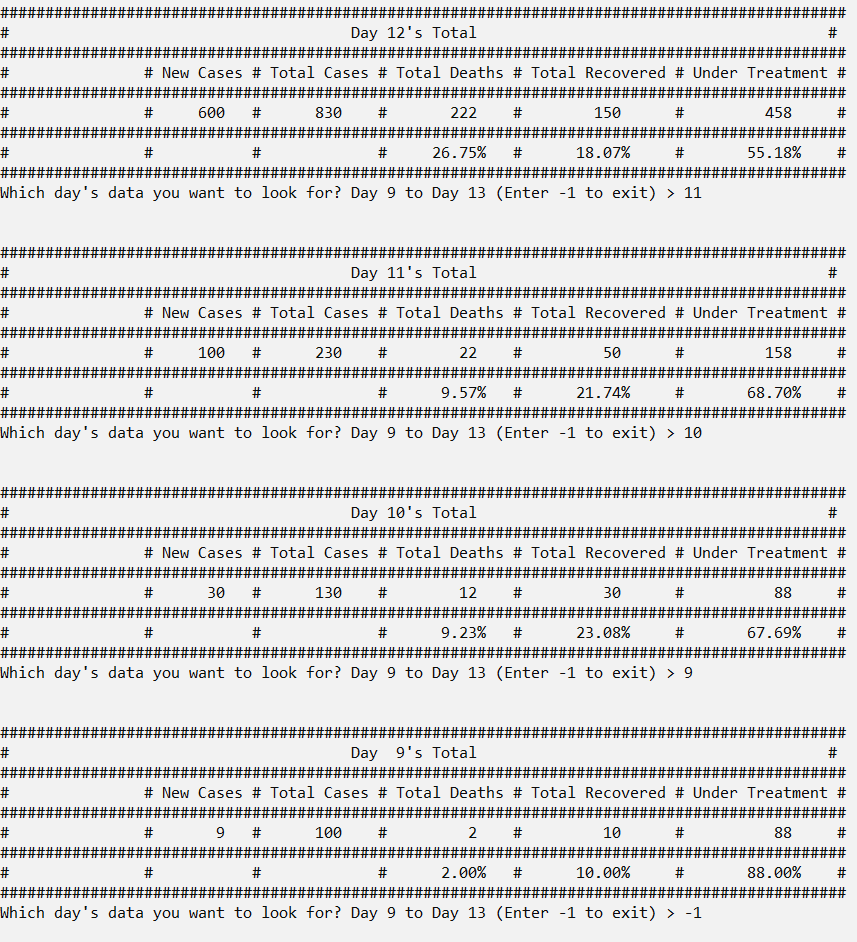
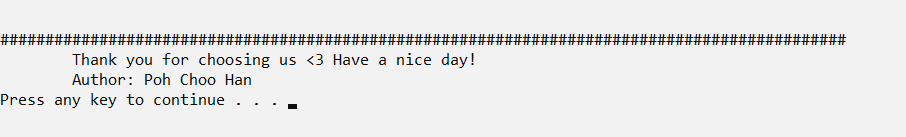
* 1. **Run 2 Scenario**

The data inputted is relatively high to test how the system deal with large amount of cases. It should show a orange warning as the grow rate suddenly increase aggressively and needed government monitor. There are total 4 days data is recorded. The highest number of cases occur at Day 13 which has 1000 new cases while the lowest number of cases is occur at Day 9 which has only 9 new cases. The grow rate rise suddenly so the system should show orange warning, which government should take some actions towards the situation. The added feature (1) user ID, name and time, (2) Average case, death, recover and warning and (5) Look back data are demonstrate and test here.

**Test Data + Expected Outputs**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Inputs | | | | Expected Results / Outputs | | | | |
| Loop No. | Day No. | New Local Cases | New Imported Cases | No. of deaths | No. recovered | New cases | Total Cases | Total Deaths | Total Recovered | Total Under Treatment |
|  | Previous  9 |  |  |  |  | 9 | 100 | 2  (2.00%) | 10 (10.00%) | 88  (88.00%) |
| 1 | 10 | 23 | 7 | 10 | 20 | 30  Increase  = 21 | 100 + 30 = 130 | 2+10=12  (9.23%)  Change = 7.23% | 10+20 = 30  (23.08%)  Change = 13.08% | 130 - 12 – 30  = 88  (67.69%) |
| 2 | 11 | 1 | 99 | 10 | 20 | 100  Increase  = 70 | 130 + 100 = 230 | 12+10 = 22  (9.57%)  Change = 0.34% | 30 + 20 = 50  (21.74%)  Change =  -1.34% | 230 – 22 – 50 = 158  (68.70%) |
| 3 | 12 | 100 | 500 | 200 | 100 | 600  Increase  = 500 | 230 + 600 = 830 | 22+200 = 222  (26.75%)  Change = 17.18% | 50 + 100 = 150  (18.07%)  Change =  -3.67% | 830 – 222 – 150 = 458  (55.18%) |
| 4 | 13 | 700 | 300 | 10 | 299 | 1000  Increase  = 400 | 830 + 1000 = 1830 | 222 + 10 = 232  (12.68%)  Change = -14.07% | 150 + 299 = 449  (24.54%)  Change =  6.47% | 1830 – 232 - 449 = 1149  (62.79%) |
|  |  |  |  |  |  | Ending message:  Total days recorded > 4 (Day 10 to Day 13)  Highest number of cases > 1000 (Occur at Day 13)  Lowest number of cases > 9 (Occur at Day 9)  Average cases = 432.50  Average deaths = 57.50  Average recover = 109.75  Yellow warning (xx), government should carry out some actions!. | | | | |



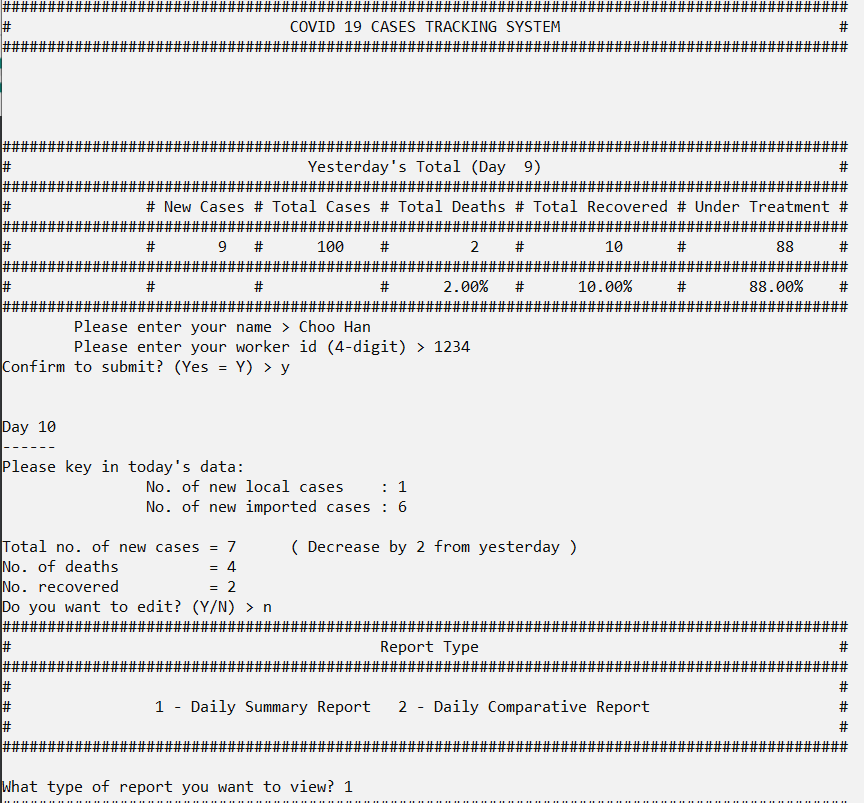
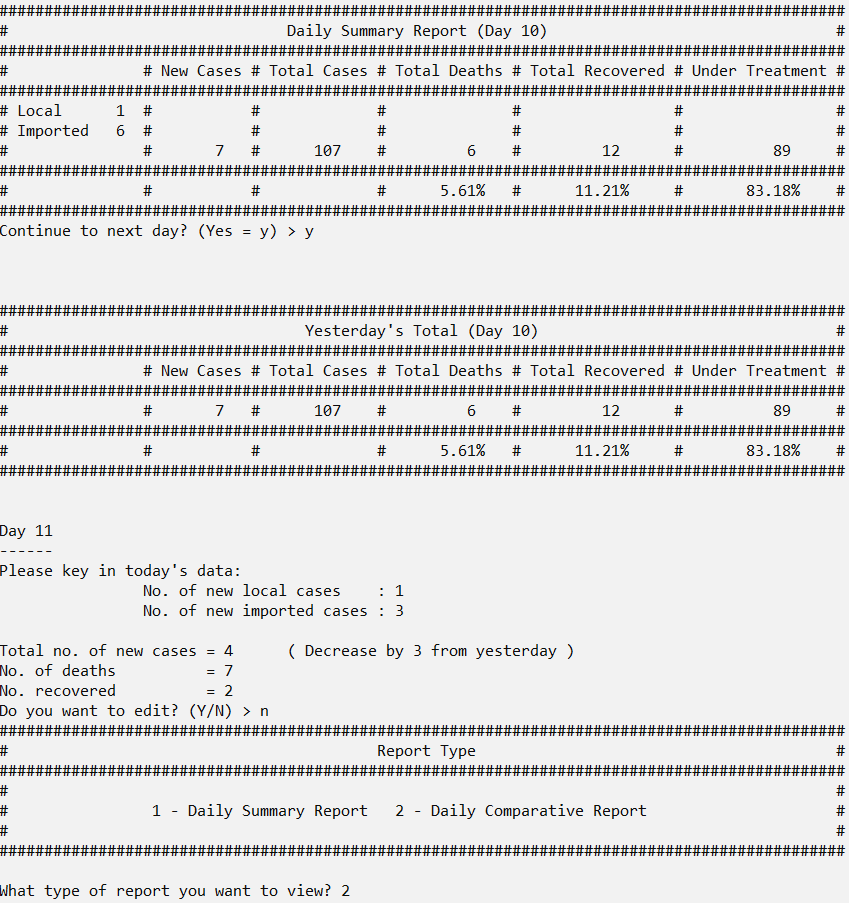
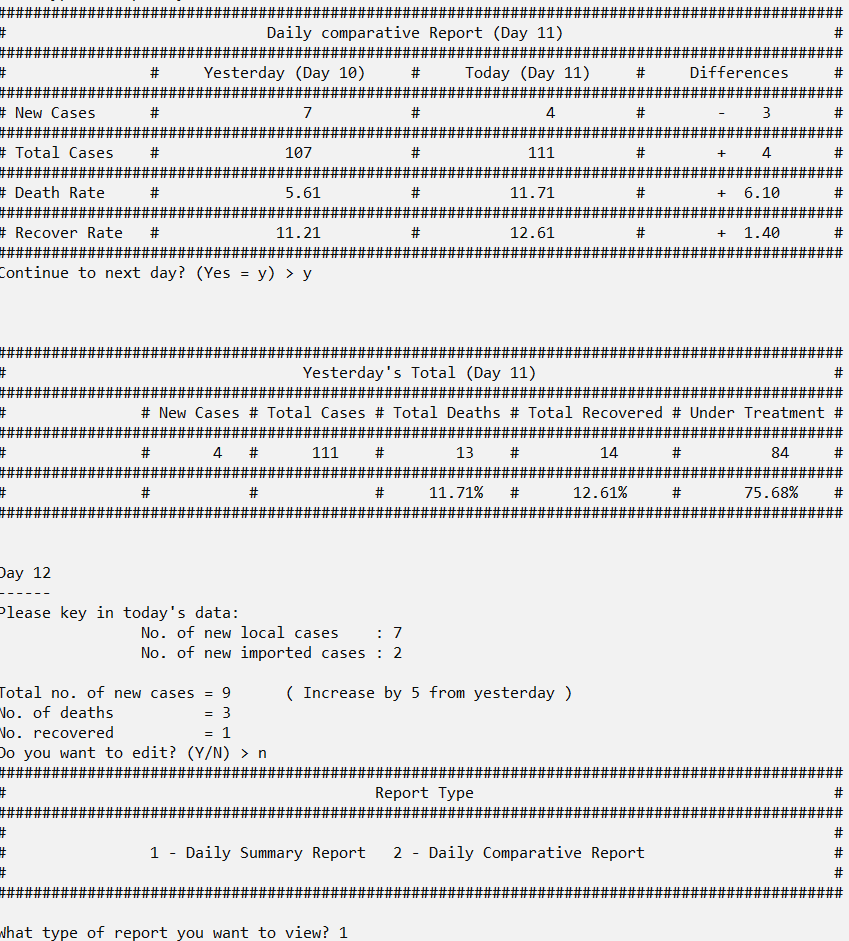
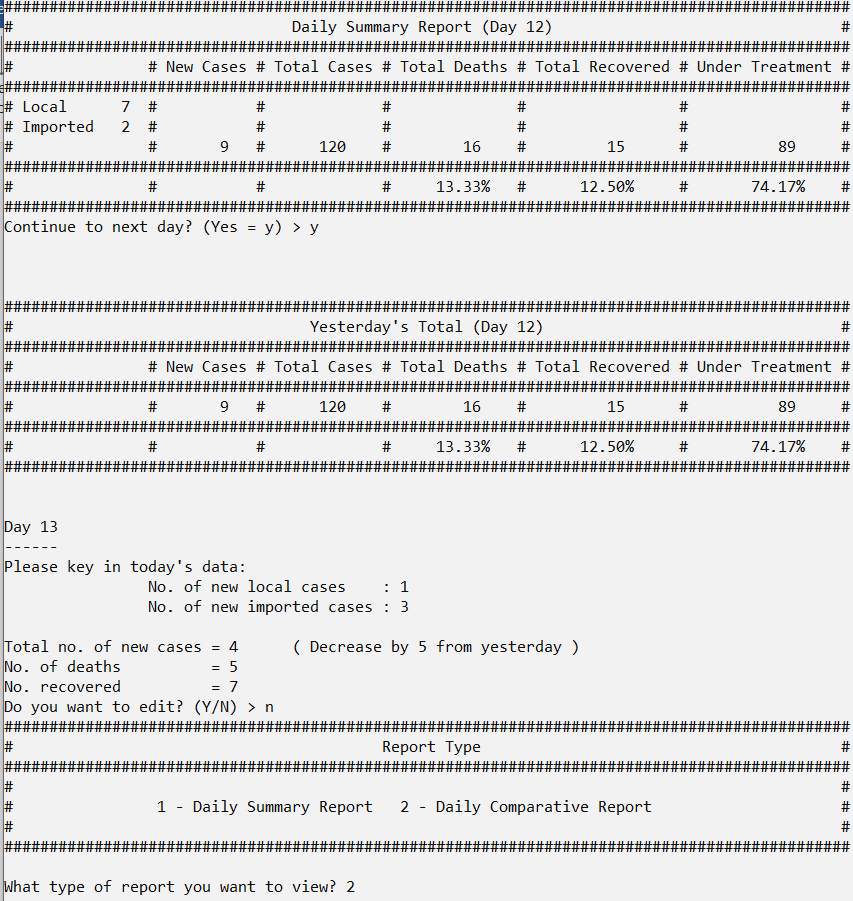
     

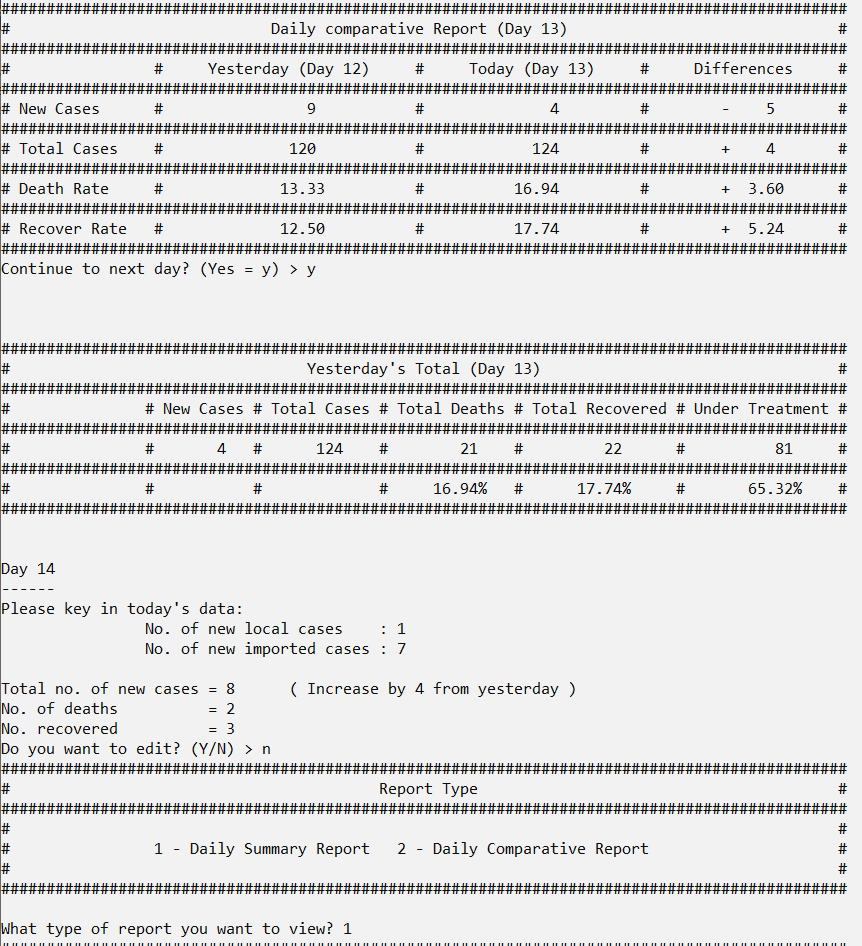
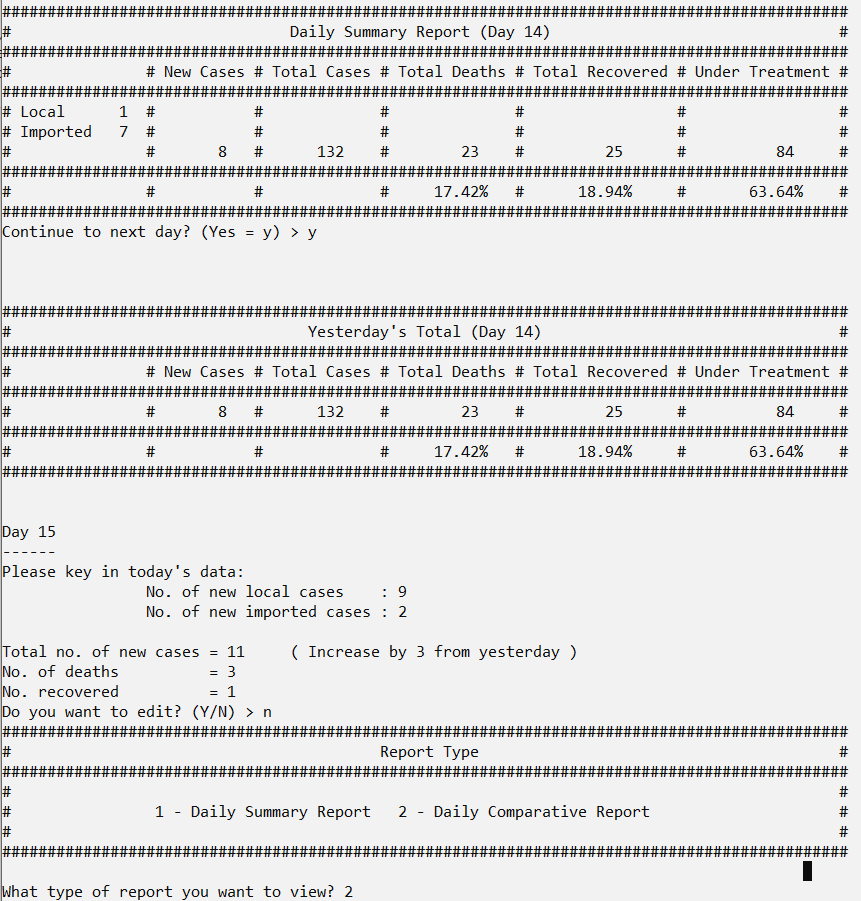
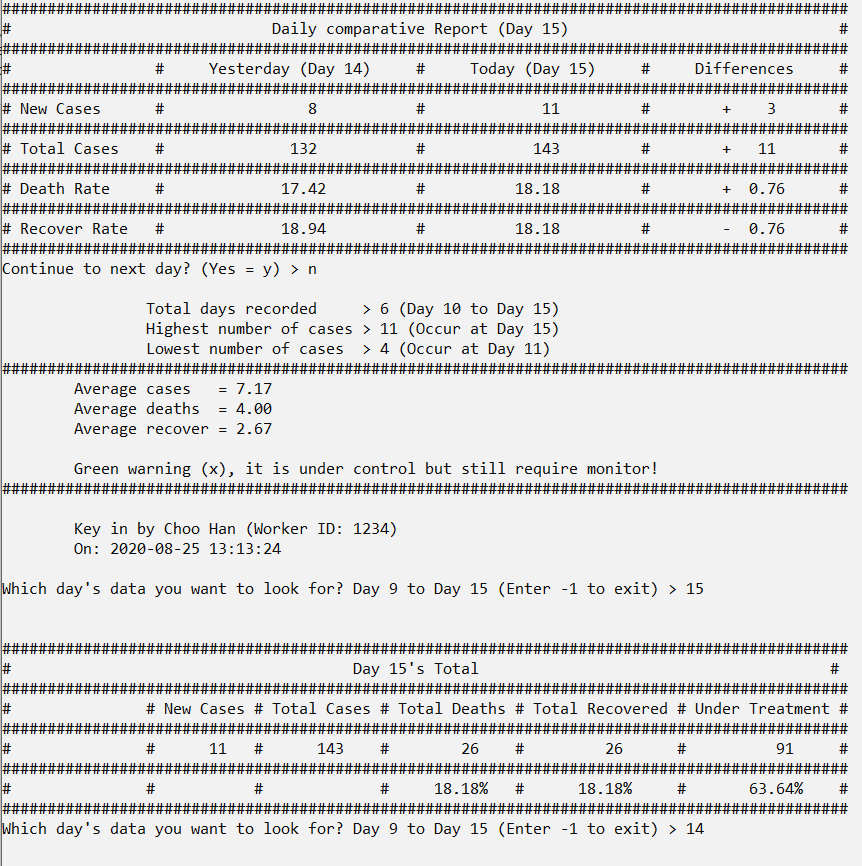
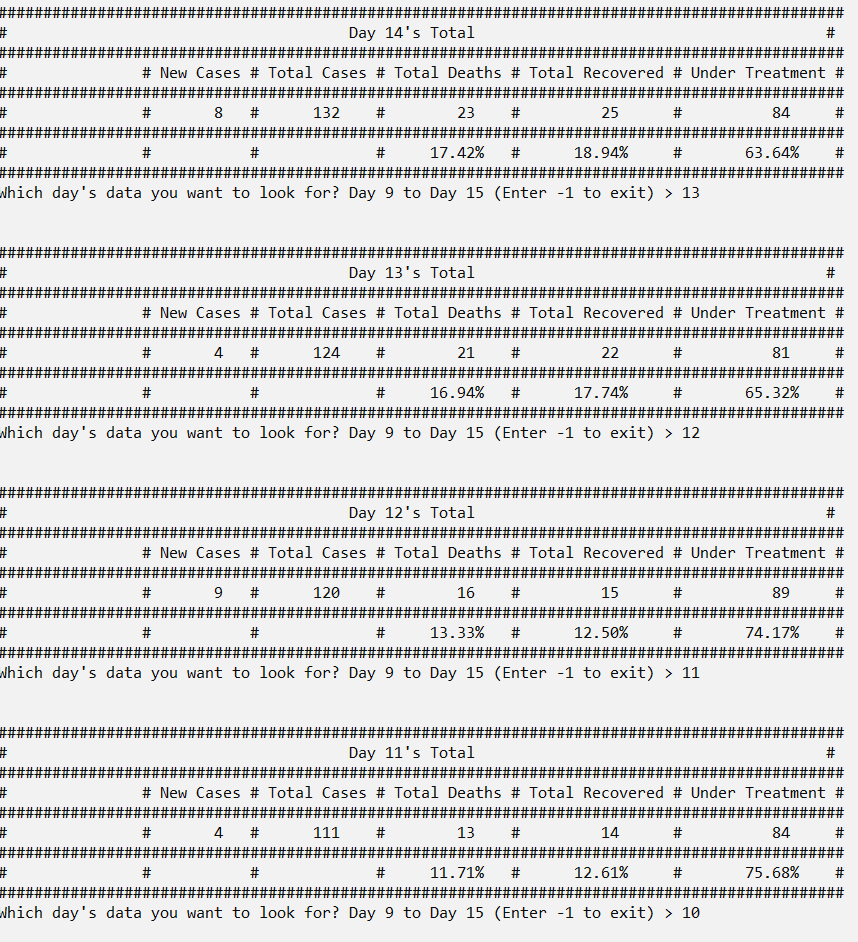
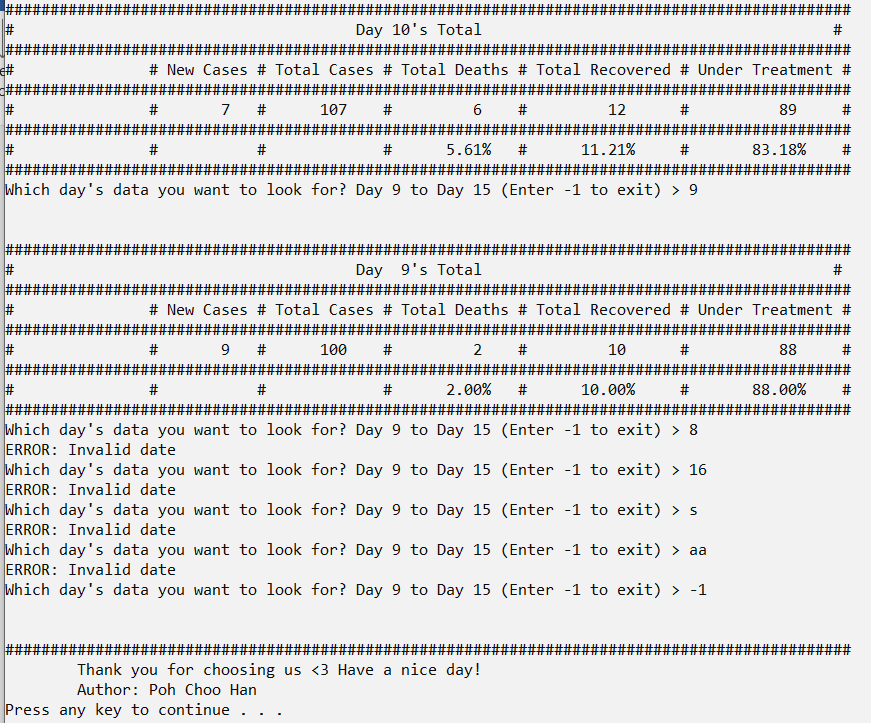
* 1. **Run 3 Scenario**

The data inputted is relatively very low to show the system can function for a extremely low data. There are total 6 days’ data is recorded to test whether the system is flexible enough to store different amount of days’ data. The highest number of cases occur at Day 15 which has 11 new cases, while the lowest number of cases is occur at Day 11 which has only 4 new cases. The grow rate is relatively very low so the system will show green warning, which mean Malaysia is still relatively safe in Covid-19 pandemic. The added feature (1) user ID, name and time, (2) Average case, death, recover and warning, (4) Validation, (5) Look back data is demonstrate and test here.

**Test Data + Expected Outputs**

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  |  | Inputs | | | | Expected Results / Outputs | | | | |
| Loop No. | Day No. | New Local Cases | New Imported Cases | No. of deaths | No. recovered | New cases | Total Cases | Total Deaths | Total Recovered | Total Under Treatment |
|  | Previous  9 |  |  |  |  | 9 | 100 | 2  (2.00%) | 10 (10.00%) | 88  (88.00%) |
| 1 | 10 | 1 | 6 | 4 | 2 | 7  Decrease  = 2 | 100 + 7 = 107 | 2+4=6  (5.61%)  Change = 3.61% | 10+2 = 12  (11.21%)  Change = 1.21% | 107 – 6 - 12  = 89  (83.18%) |
| 2 | 11 | 1 | 3 | 7 | 2 | 4  Decrease  = 3 | 107 + 4 = 111 | 6+7 = 13  (11.71%)  Change = 6.1% | 12+2 = 14  (12.61%)  Change =  1.4% | 111 – 13 - 14= 84  (75.68%) |
| 3 | 12 | 7 | 2 | 3 | 1 | 9  Increase  = 5 | 111 + 9 = 120 | 13+3 = 16  (13.33%)  Change = 1.62% | 14+1 = 15  (12.5%)  Change =  -0.11% | 120 – 16 - 15 = 89  (74.17%) |
| 4 | 13 | 1 | 3 | 5 | 7 | 4  Decrease  = 5 | 120 + 4 = 124 | 16 + 5 = 21  (16.94%)  Change = 3.61% | 15 + 7 = 22  (17.74%)  Change =  5.24% | 124 – 21 - 22 = 81  (65.32%) |
| 5 | 14 | 1 | 7 | 2 | 3 | 8  Increase  = 4 | 124 + 8 = 132 | 21 + 2 = 23  (17.42%)  Change = 0.48% | 22 + 3 = 25  (18.94%)  Change =  1.2% | 132 – 23 - 25 = 84  (63.64%) |
| 6 | 15 | 9 | 2 | 3 | 1 | 11  Increase  = 3 | 132 + 11 = 143 | 23 + 3 = 26  (18.18%)  Change = 0.76% | 25 + 1 = 26  (18.18%)  Change =  -0.76% | 143 – 26 - 26 = 91  (63.64%) |
|  |  |  |  |  |  | Ending message:  Total days recorded > 6 (Day 10 to Day 15)  Highest number of cases > 11 (Occur at Day 15)  Lowest number of cases > 4 (Occur at Day 11)  Average cases = 7.17  Average deaths = 4.00  Average recover = 2.67  Green warning (x), it is under control but still require monitor! | | | | |

**6.0 Constants & variables**

|  |
| --- |
| **Pre processor**  #define PNEW 9  #define PTOTAL 100  #define PDEATH 2  #define PRECOVER 10  #define PTREATMENT 88  int days = 9;  dailyCases dc[365];  **Main function**  int todayLocalCase, todayImportCase, todayDeath, todayRecover;  int currentTotal, currentDeath, currentRecover;  int yesterdayNew, yesterdayTotal, yesterdayDeath, yesterdayRecover;  int todayTotal;  int highCase, highDay, lowCase, lowDay;  char editC;  char name[50];  int workerID;  char nameCon;  int choice;  char con;  char temp;  int difference = todayTotal - yesterdayNew;  int difference = yesterdayNew - todayTotal;  float dayAmount = days - 9;  float average = ((float)currentTotal - (float)PTOTAL) / dayAmount;  float deathAverage = ((float)currentDeath - (float)PDEATH) / dayAmount;  float recoverAverage = ((float)currentRecover - (float)PRECOVER) / dayAmount;  time\_t t = time(NULL);  struct tm tm = \*localtime(&t);  **header function**  int underTreatment = currentTotal - currentDeath - currentRecover;  float deathRate = (float)PDEATH / (float)PTOTAL \* 100.0;  float recoverRate = (float)PRECOVER / (float)PTOTAL \* 100.0;  float utRate = (float)PTREATMENT / (float)PTOTAL \* 100.0;  **getNewLocalCases & getNewImportCases & getNewDeath & getNewRecover function**  int n;  char temp;  **printSummary function**  int underTreatment = currentTotal - currentDeath - currentRecover;  float deathRate = (float)currentDeath / (float)currentTotal \* 100.0;  float recoverRate = (float)currentRecover / (float)currentTotal \* 100.0;  float utRate = (float)underTreatment / (float)currentTotal \* 100.0;  **printCompare function**  float ytdDeathRate = (float)ytdDeath / (float)ytdCurrentTotal \* 100.0;  float ytdRecoverRate = (float)ytdRecover / (float)ytdCurrentTotal \* 100.0;  float tdyDeathRate = (float)todayDeath / (float)todayCurrentTotal \* 100.0;  float tdyRecoverRate = (float)tdyRecover / (float)todayCurrentTotal \* 100.0;  **checkInt function**  char test[50];  int length = strlen(test);  **record function**  int index = days - 9;  **lookback function**  int index = choice - 9; |

**Constants**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Constant** | **Values** | **Purpose** |
| 1. | PNEW | 9 | It stores the number of new cases on Day 9. |
| 2. | PTOTAL | 100 | It stores balance of the total cases that are recorded on previous days which are before the user run the program. |
| 3. | PDEATH | 2 | It stores balance of the total number of deaths that are recorded on previous days which are before the user run the program. |
| 4. | PRECOVER | 10 | It stores balance of the total number of people recovered that are recorded on previous days which are before the user run the program. |
| 5. | PTREATMENT | 88 | It stores balance of the total number of people under treatment that are recorded on previous days which are before the user run the program. |

**Variables**

|  |  |  |  |
| --- | --- | --- | --- |
|  | **Variable** | **Data Type** | **Purpose** |
| 1. | todayLocalCase | Integer | Use to store the total local case that is input by the user in scanf function (return from getNewLocalCases function). |
| 2. | todayImportCase | Integer | Use to store the total import case that is input by the user in scanf function (return from getNewImportCases function). |
| 3. | todayTotal | Integer | Use to store the result after the summation of todayLocalCase and todayImportCase to find total new case of the day. |
| 4. | todayDeath | Integer | Use to store the total number of deaths that is input by the user in scanf function (return from getNewDeath function). |
| 5. | todayRecover | Integer | Use to store the total number of people who recovered that is input by the user in scanf function (return from getNewRecover function). |
| 6. | currentTotal | Integer | Use to store the current total cases balance, which is the result of adding the yesterdayTotal and todayTotal. |
| 7. | currentDeath | Integer | Use to store the current total death balance, which is the result of adding the yesterdayDeath and todayDeath. |
| 8. | currentRecover | Integer | Use to store the current total number of people recovered balance, which is the result of adding the yesterdayRecover and todayRecover. |
| 9. | underTreatment | Integer | Use to store the current number of people under treatment which is the result of currentTotal minus currentDeath minus currentRecover. |
| 10. | deathRate | Float | Use to store the death rate percentage which is the result of currentDeath divide by currentTotal and multiply 100.0. It is use in header function and printSummary function |
| 11. | recoverRate | Float | Use to store the recover rate percentage which is the result of currentRecover divide by currentTotal and multiply 100.0. It is use in header function and printSummary function. |
| 12. | utRate | Float | Use to store the under treatment rate percentage which is the result of underTreatment divide by currentTotal and multiply 100.0. It is use in printSummary function only. |
| 13. | days | Integer | Use to store and update current day so that system is keep tracking and update of the day status. |
| 14. | yesterdayNew | Integer | Use to store previous day’s new cases before start storing new day’s new cases. |
| 15. | yesterdayTotal | Integer | Use to store previous day’s cases total balance before start storing new day’s cases total balance. |
| 16. | yesterdayDeath | Integer | Use to store previous day’s total people death balance before start storing new day’s total death balance. |
| 17. | yesterdayRecover | Integer | Use to store previous day’s total recovered victim balance before start storing new day’s total recovered victim balance. |
| 18. | highCase | Integer | Use to store and update the highest amount of cases being recorded. |
| 19. | highDay | Integer | Use to store and update the day which has the highest amount of cases begin recorded |
| 20. | lowCase | Integer | Use to store and update the lowest amount of cases being recorded |
| 21. | lowDay | Integer | Use to store and update the day which has the lowest amount of cases being recorded. |
| 22. | editc | Character | Use to store and check the user’s response towards choice on whether edit the inputted data. |
| 23. | Name[50] | Character (Array) | Use to store the user’s name. |
| 24. | workerID | Integer | Use to store the user’s worker ID. |
| 25. | nameCon | Character | Use to store and check the user’s responses towards choice on whether submit the user’s name and user’s worker ID. |
| 26. | choice | Integer | Use to store the user’s response |
| 27. | con | Character | Use to store and check the user’s response towards choice on reports type to print. |
| 28. | difference | Integer | Use to store the answer of the substraction between yesterdayNew and todayTotal. (The larger value variable minus the smaller value variable) |
| 29. | dayAmount | Float | Use to calculate the amount of day recorded and store it in this variable. (days minus 9) |
| 30. | average | Float | Use to count the average cases over the recorded day and store it in this variable  (currentTotal minus PTOTAL and divide by dayAmount) |
| 31. | deathAverage | Float | Use to count the average death over the recorded day and store it in this variable. (currentDeath minus PDEATH and divide by dayAmount) |
| 32. | recoverAverage | Float | Use to count the average recover over the recorded day and store it in this variable.  (currentRecover minus PRECOVER and divide by dayAmount) |
| 33. | t | Time\_t | Use to obtain current time and date and store it inside this variable. |
| 34. | tm | struct | Use to store the current date and time (t) into struct so that it is easier to obtain the details. |
| 35. | n | Integer | Act as temporarily integer storages for the data key in by the user before return to the main function. |
| 36. | ytdDeathRate | Float | Use to count the yesterday death rate.  ytdDeath (yesterdayDeath) divide by ytdCurrentTotal (yesterdayTotal) and multiply by 100.0 |
| 37. | ytdRecoverRate | Float | Use to count the yesterday recover rate.  ytdRecover(yesterdayRecover) divide by ytdCurrentTtotal (yesterdayTotal) and multiply by 100.0 |
| 38. | tdyDeathRate | Float | Use to count the today death rate  tdyDeath (currentDeath) divide by tdyCurrentTotal (currentTotal) and multiply by 100.0 |
| 39. | tdyRecoverRate | Float | Use to count the today recover rate  tdyRecover (currentRecover) divide by tdyCurrentTotal (currentTotal) and multiply by 100.0 |
| 40. | Test[50] | Character (Array) | Act as a temporarily storage while passing the integer to this function. The integer passing into the function is stored inside this variable which convert it to string variable. |
| 41. | length | int | Use to store the size of the Test variable. |
| 42. | Temp | char | Use to store and check is there a space or another value is key in after the number. |
| 43. | Index | int | Use to determine the location that the data should store inside the array. |
| 44. | dc[] | dailyCases | dailyCases is a structure type that I defined to store daily cases. dc is the array that I create based on the dailyCases structure and store every day’s cases. |