LCS_PCOM7E June 2022

Launching into Computer Science

Assignment 1 – Part 1

The application is a phone book named as “My Phone Book” for facilitate user to locate the contact numbers.

My Phone Book uses dialog-like interface to communicate with the user so the users do not need to remember any command. When program starts, a service selection dialog enable the user to choose a service. The services available are (a) insert, (b) delete, (c) modify, (d) search, and (e) sort. To be user friendly, user inputs are case insensitive. If the user other chartered are keyed in, error message will appear and the program will go back to the original question.

When it comes to the delete or modify a record, a confirmation question will be asked in order to provide the user a chance to confirm their choice. When the service or program ends, a closing message will appear.

Design of data structure

There are three data the user has to input, which are name, telephone number and mobile number, to three database. Data type of all input is string. The database are recorded using the list operation. When writing simple program, it may be more efficient to use user written code instead of using library code (Jarvis, 2016).

When a record is added, each attribute is appended to the end of respective list. A graphical illustration of the data structure is as follows:

Name List	
1.	Anna
2.	Elsa
3.	Kristoff
:	
:	

Number List	
1.	Anna's phone no.
2.	Elsa's phone no.
3.	Kristoff's phone no.
:	
:	

Mobile List	
1.	Anna's mobile no.
2.	Elsa's mobile no.
3.	Kristoff's mobile no.
:	
:	

The attributes of the same record will be entered to the same index in their respective list. Their positions form the key for the program to locate the other corresponding attributes of the same record when performing other services of the application.

Algorithms of each service

The attached flowchart details the algorithms of each service. The following are narrative to the flowchart for summarising algorithms.

The program starts with the choice of service: (a) insert (b) delete (c) amend (d) search or (e) sort. Each choice represents running a separate function in the program. If

1. Insert

Insert service starts by entering a new name. The program will check if the name is duplicated with the existing names. If not, other attributes are to be entered. As each attributes of a new record is appended to the end of their respective list at the same time, therefore they have the same index in the respective list. The position forms the key for the other services of the application.

If duplication is detected, error message will be printed and service ends too.

2. Delete

Delete service starts by choosing to find the record to be deleted by name, telephone number or mobile number, and enter the name, telephone number or mobile according to the choice. All record in the database is compared to the input. If it is found, index of the attributes is checked and all other attributes of the same record are located by using the same index of the list and the whole record is identified. User is asked to reconfirm by entering yes or no. If yes, the deleted record will be printed and service ends. If no, service ends.

If other character is keyed, print error message and user has to re-confirm again.

If record not found, print error message and request user to input again.

3. Modify

Modify service starts by entering the name of record to be modified. The input is compared to the name list and index of the input in the name list is found. For the found name, choose attribute to modify and enter the new data. The Users are asked to reconfirm by entering yes or no. If yes, the record to be modified is located by the index found and being replacing by the new data. Thereafter the modified record will be printed and service ends. If no, service ends. If other character is keyed, print error message and user has to re-confirm.

If record not founded, print error message and request user to input again.

4. Search

Search service starts by choosing attribute to search and then enter the data to search. Input is compare to the database in the correct list and found the index of the matched data. Other attributes of the same index in other list are also pull out and the whole record is printed out.

If data entered is not found, goes back to re-enter. If wrong charter is entered in selecting search attributes, print error message and goes back to enter again.

5. Sort

Sort service starts by choosing attributes to sort with (e.g. sorted by name). A new (name) list will be created from the original list. Then the (name) name list will be sorted in ascending or descending order as choose by user. For each record in the new name

list, their index in the original name list is found. When print the new name list, the element in the other list with the same index found will be printed.

If other characters are keyed when selecting the attributes or order, error message will be printed and goes back to choosing attributes or order.

6. Closing

When the above services ends, “thank you” message will be printed. Then the user will be asked if he want to start a new service or end service. If start a new service, go to beginning of the program. If end service, closing messing will be printed and program ends.

Test plan

The test plans for the program and each function are set out in the table below.

Test data used:

Name: Anna, Telephone: 111, Mobile: 1111

Name: Elsa, Telephone: 222, Mobile: 2222

Name: Kristoff, Telephone: 333, Mobile: 3333

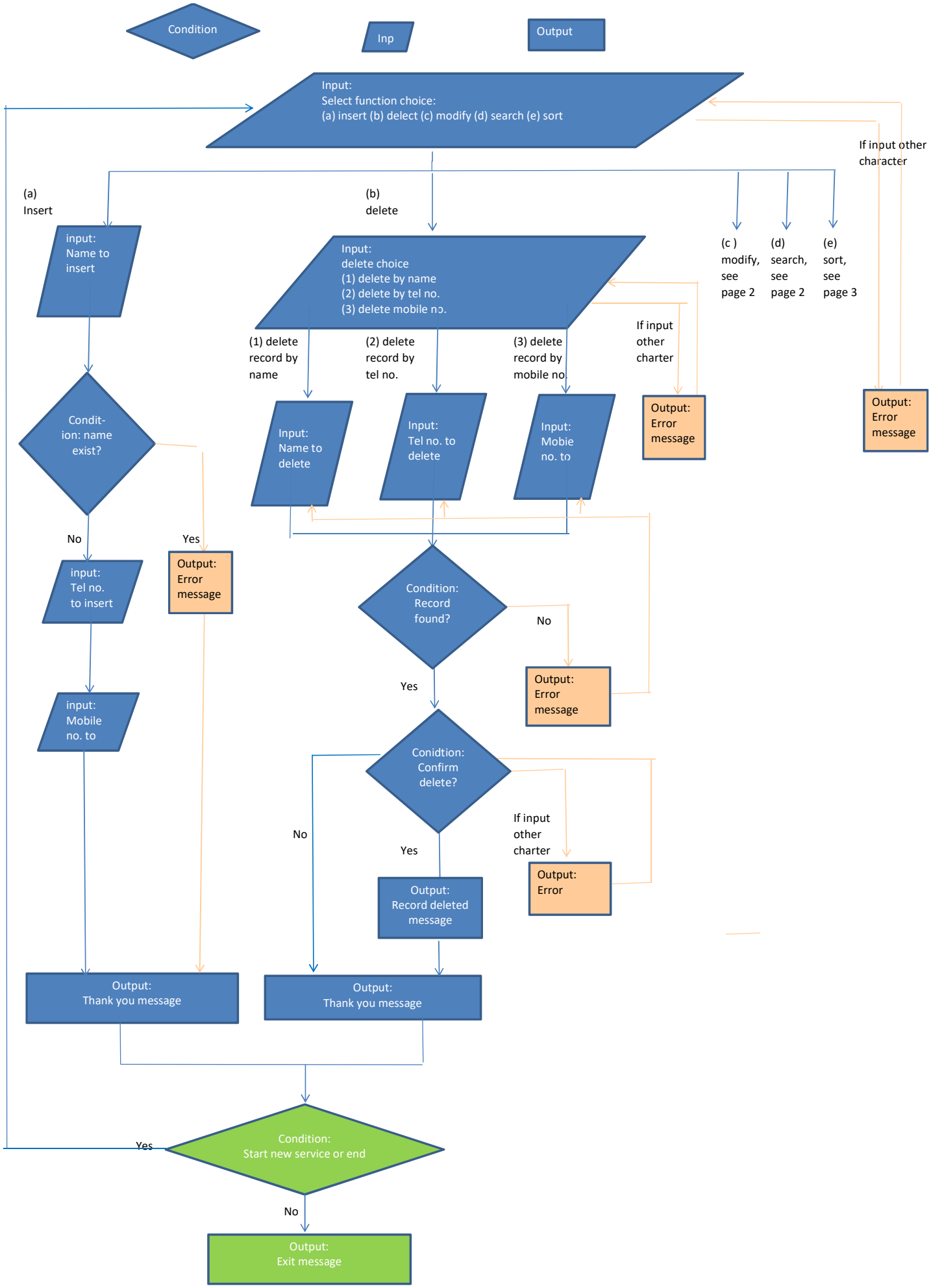
Function	**Step to be tested**	**Testing procedure**		**Expected result**
1.Selet service	1.1 Service choice	Check case sensitivity		Case insensitive
		Key other character to service choice		Error message and go back to service choice
2. Insert	2.1 Enter data	Enter the test data		Able to print the three database (lists) at the end of the function
	2.2 Duplicated name	Enter a duplicated name		Error message and function ends
3. Delete	3.1 Delete choice	Enter existing name and key other character for delete choice		Error message and goes back to delete choice
	3.2 Delete function	For each choice	Cofirm delete ?	Result
		Enter delete field, then	“yes” (different case)	Print out record changed (case insensitive)
			“no” (different case)	Closing message (case insensitive)
			enter “others”	Error message, goes back to re-confirm
		Enter non-exist data		Error message, goes back to re-enter

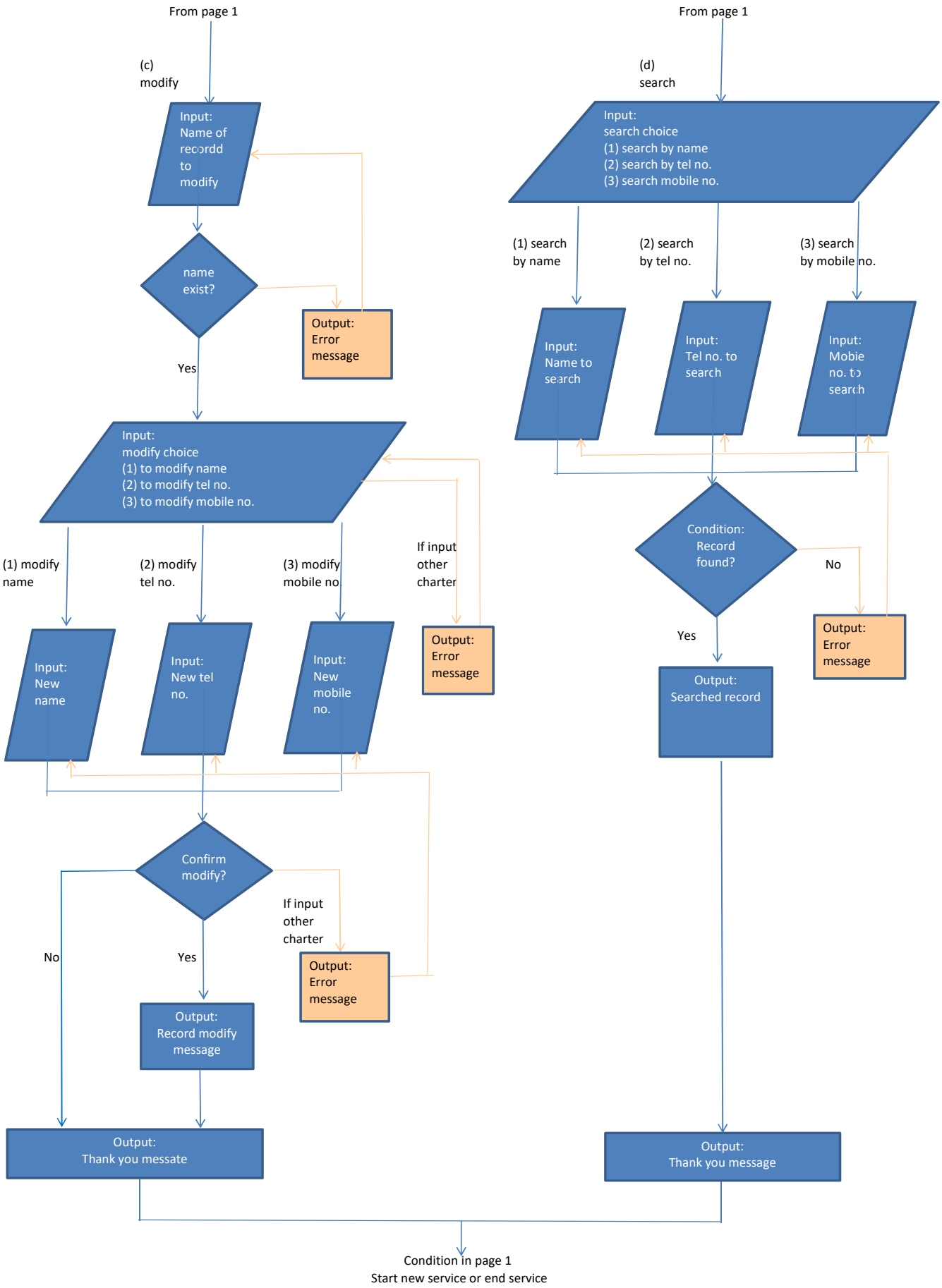
Function	Step to be tested	Testing procedure	Expected result
4. Modify	4.1 Modify name	Enter non-exist name	Error message, goes back to re-enter
	4.2 Modify choice	Enter existing name and key other character for modify choice	Error message, goes back to modify choice
	4.3 Modify function	Same as 3.2 (enter new attribute to be modified)	Same as 3.2
5. Search	5.1 Search choice	Key other character	Error message, goes back to modify choice
	5.2 Search function	Enter attribute to search (e.g. Anna)	If record found, print search record (e.g. whole record of Anna), if not found, go back to re-enter
6. Sort	6.1 Sort choice	Key other character	Error message, goes back to sort or order choice
	6.2 Order choice		
	6.1 & 6.2 Sort choice and order choice	Enter correct attribute to sort	Print correct list of correct order

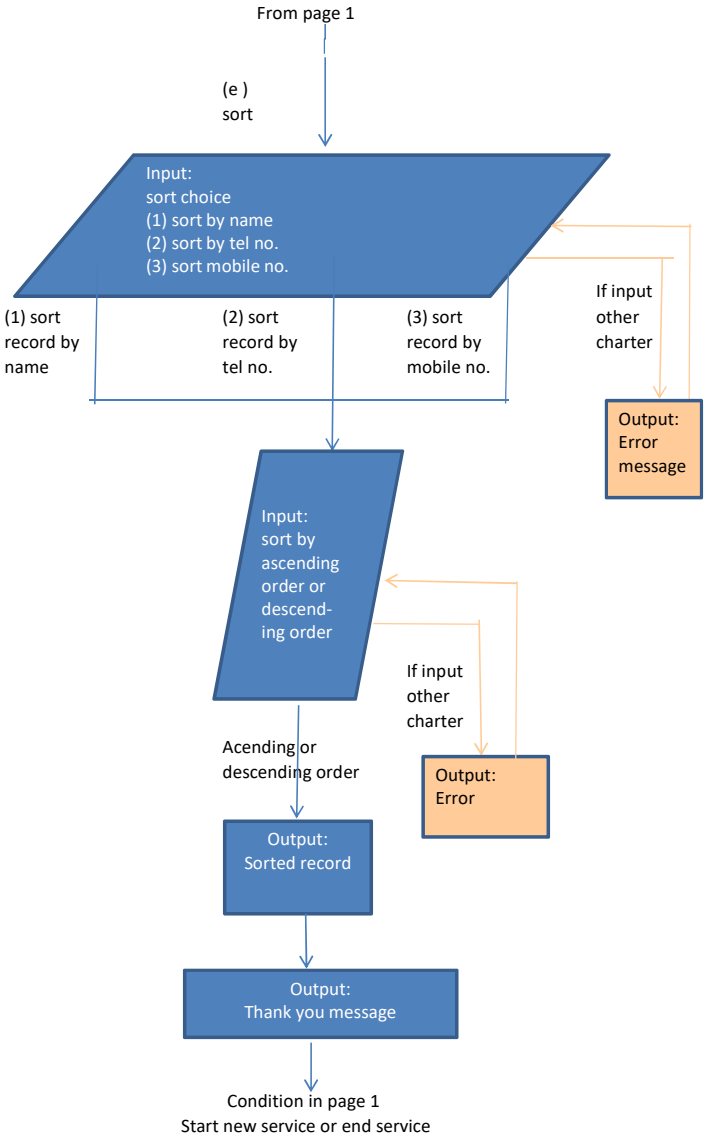
Function	Step to be tested	Testing procedure	Expected result
7. Closing	After all above service	Start service	Goes back to 1
		End service	Print closing message

Reference:

Jarvis, D. (2016) How do you decide whether to use a library or write your own implementation. Available from: https://stackoverflow.com/questions/1236256/how-do-you-decide-whether-to-use-a-library-or-write-your-own-implementation [Accessed 8 August 2022]


Condition
Inp
Output
Input:
Select function choice:
(a) insert (b) delect (c) modify (d) search (e) sort
If input other character
(a)
Insert
(b)
delete
input:
Name to insert
Input:
delete choice
(1) delete by name
(2) delete by tel no.
(3) delete mobile no.
(c) modify, see page 2
(d) search, see page 2
(e) sort, see page 3
If input other charter
(1) delete record by name
(2) delete record by tel no.
(3) delete record by mobile no
Output:
Error message
Output:
Error message
Condit-
ion: name
exist?
Input:
Name to
delete
Input:
Tel no. to
delete
Input:
Mobie
no. to
No
Yes
input:
Tel no.
to insert
Output:
Error
message
Condition:
Record
found?
No
Output:
Error
message
Yes
input:
Mobile
no. to
Conidtion:
Confirm
delete?
No
Yes
If input
other
charter
Output:
Error
Output:
Record deleted
message
Output:
Thank you message
Output:
Thank you message
Condition:
Start new service or end
Yes
No
Output:
Exit message


From page 1
(c)
modify
Input:
Name of
recordd
to
modify
name
exist?
Output:
Error
message
Yes
Input:
modify choice
(1) to modify name
(2) to modify tel no.
(3) to modify mobile no.
(1) modify
name
(2) modify
tel no.
(3) modify
mobile no
If input
other
charter
Input:
New
name
Input:
New tel
no.
Input:
New
mobile
no.
Output:
Error
message
Confirm
modify?
No
Yes
If input
other
charter
Output:
Error
message
Output:
Record modify
message
Output:
Thank you messate
From page 1
(d)
search
Input:
search choice
(1) search by name
(2) search by tel no.
(3) search mobile no.
(1) search
by name
(2) search
by tel no.
(3) search
by mobile no.
Input:
Name to
search
Input:
Tel no. to
search
Input:
Mobie
no. to
search
Condition:
Record
found?
No
Yes
Output:
Error
message
Output:
Searched record
Output:
Thank you message
Condition in page 1
Start new service or end service


From page 1
(e)
sort
Input:
sort choice
(1) sort by name
(2) sort by tel no.
(3) sort mobile no.
(1) sort record by name
(2) sort record by tel no.
(3) sort record by mobile no.
If input other charter
Output: Error message
Input: sort by ascending order or descend- ing order
If input other charter
Acending or descending order
Output: Error
Output: Sorted record
Output: Thank you message
Condition in page 1
Start new service or end service