## Class Activity – Contacts Arraylist

Design a **Contact** class that the following members:

- A field for the name of the person (a string).
- A field for the cell phone number of the person (a string).

Constructors: A default constructor and a parameterized constructor.

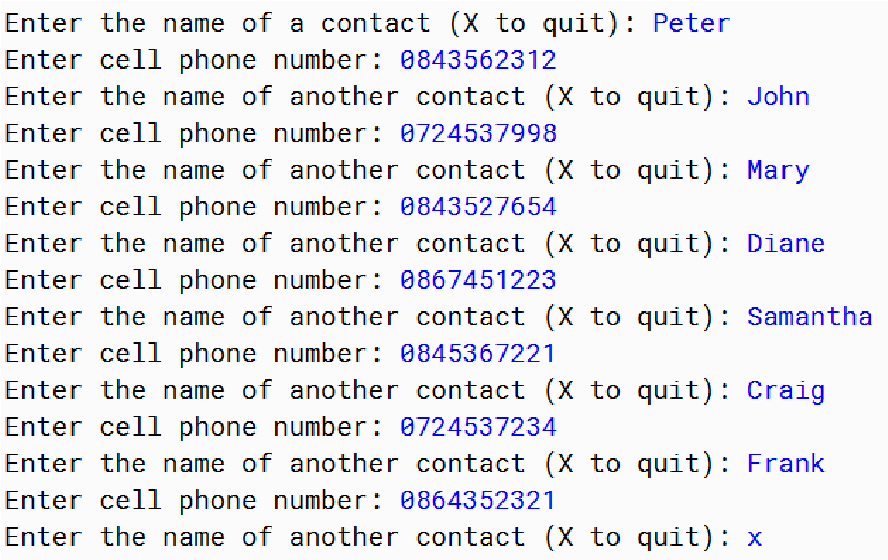
- Accessors methods for the data fields.
- Set methods for the data fields.

- A **toString()** method that compiles a string that is formatted to display the data fields.

Design a **testContacts** driver class that has a **lstContacts** arraylist that can hold up to a maximum of 15 objects.

Write a static **addContacts()** method that receives the **lstContacts** arraylist and allow the user to enter the information of a number of contacts. If the answer on whether to add another contact is N, terminate the loop and return the arraylist to the call statement.

Example of input:

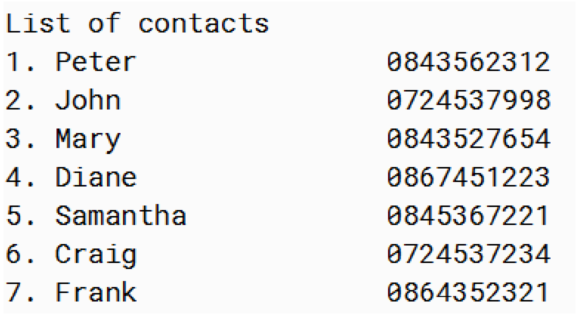


```
Enter the name of a contact (X to quit): Peter
Enter cell phone number: 0843562312
Enter the name of another contact (X to quit): John
Enter cell phone number: 0724537998
Enter the name of another contact (X to quit): Mary
Enter cell phone number: 0843527654
Enter the name of another contact (X to quit): Diane
Enter cell phone number: 0867451223
Enter the name of another contact (X to quit): Samantha
Enter cell phone number: 0845367221
Enter the name of another contact (X to quit): Craig
Enter cell phone number: 0724537234
Enter the name of another contact (X to quit): Frank
Enter cell phone number: 0864352321
Enter the name of another contact (X to quit): x
```

Write a static **displayContacts()** method that receives the **lstContacts** arraylist as parameter and display a numbered list of information.

Call the **displayContacts()** method in the main method to display the list.

Example of output:



```
List of contacts
1. Peter                0843562312
2. John                 0724537998
3. Mary                 0843527654
4. Diane                0867451223
5. Samantha             0845367221
6. Craig                0724537234
7. Frank                0864352321
```

Write a static method called **findAContact()** that receives the **lstContacts** arraylist.
Ask the user to enter the name of the contact.
If the name is in the arraylist, display a suitable message which includes the name and cell number of the contact. Ask whether the user wants to change the cell phone number. If the answer is Yes, ask the user to enter the new number and use the **setCellNum()** method to set the contact person's cell number to the new number.
Display the list of contacts.

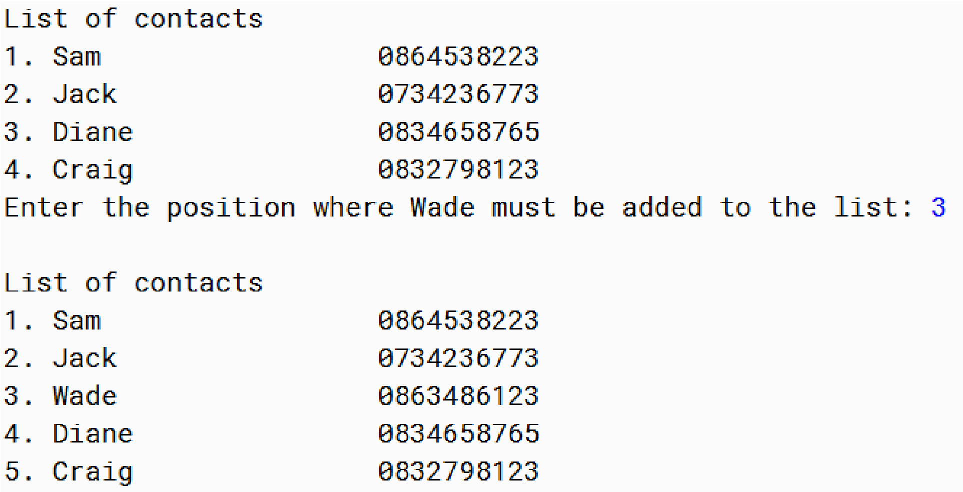
If the name is not on the list, ask the user whether the name should be added to the list. If the answer is Yes, ask the user to enter the person's cell number.
**NOTE:**
The name that has been entered must by part of the prompt messages to the user.
Display the current list (without the new contact) using the **displayContacts()** method and ask the user to enter the position where the new contact should be added into the list.
Validate the number that is entered. It should not be a negative number and should not exceeding the size of the arraylist. Repetitively ask for a number to be entered until the number is valid.
Insert the contact in the correct position in the arraylist and call the **displayContacts()** method to display the list.

Example run 1:

```
Enter the name of a contact (X to quit): James
Enter cell phone number: 0834568234
Enter the name of another contact (X to quit): Peter
Enter cell phone number: 0835492440
Enter the name of another contact (X to quit): Sam
Enter cell phone number: 0835238550
Enter the name of another contact (X to quit): x

Enter the name of a contact to lookup: Peter

Peter is on the list.
The cell number is 0835492440
Do you want to change the cell number (Y or N)?: Y

Enter the new cell number: 0845346221

List of contacts
1. James               0834568234
2. Peter               0845346221
3. Sam                 0835238550
```

Example run 2:

```
Enter the name of a contact (X to quit): Sam
Enter cell phone number: 0864538223
Enter the name of another contact (X to quit): Jack
Enter cell phone number: 0734236773
Enter the name of another contact (X to quit): Diane
Enter cell phone number: 0834658765
Enter the name of another contact (X to quit): Craig
Enter cell phone number: 0832798123
Enter the name of another contact (X to quit): x

Enter the name of a contact to lookup: Wade

Wade is not on the list
Do you want to add the contact to the list (Y or N)?: y

Enter Wade's cell number: 0863486123

Current list
```

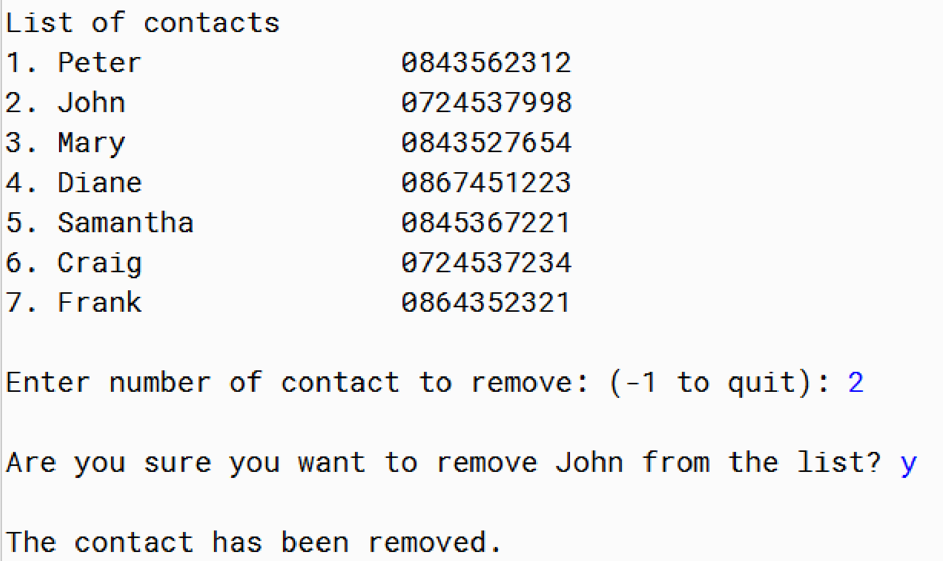


```
List of contacts
1. Sam                  0864538223
2. Jack                 0734236773
3. Diane                0834658765
4. Craig                0832798123
Enter the position where Wade must be added to the list: 3

List of contacts
1. Sam                  0864538223
2. Jack                 0734236773
3. Wade                 0863486123
4. Diane                0834658765
5. Craig                0832798123
```

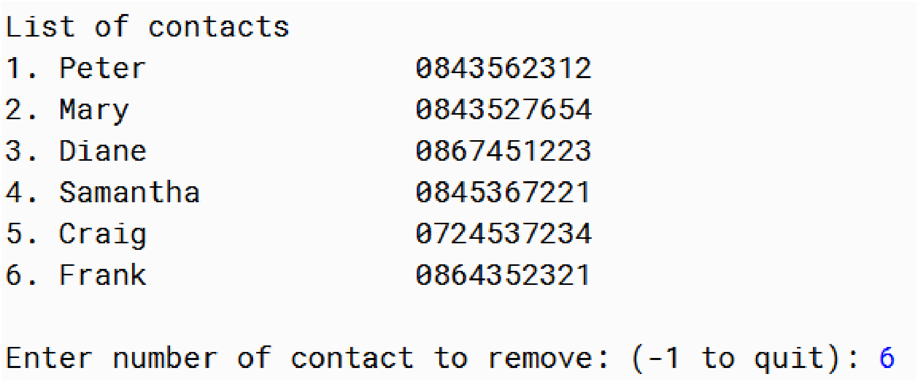
Write a static method called **removeContacts()** that receives the **lstContacts** arraylist.
Call the **displayContacts()** method to display the numbered list of contacts. Ask the user to enter a number from the list (contact to be removed) or -1 to quit. Ask whether the user is sure about removing the contact from the list. On confirmation, remove the contact from the arraylist and display the list again.
Repeat the process until -1 is entered or the arraylist is empty. If the the arraylist is empty, display a suitable message and terminate the process of removing contacts from the list.

Example runs of the **removeContacts()** method with data that has been entered.

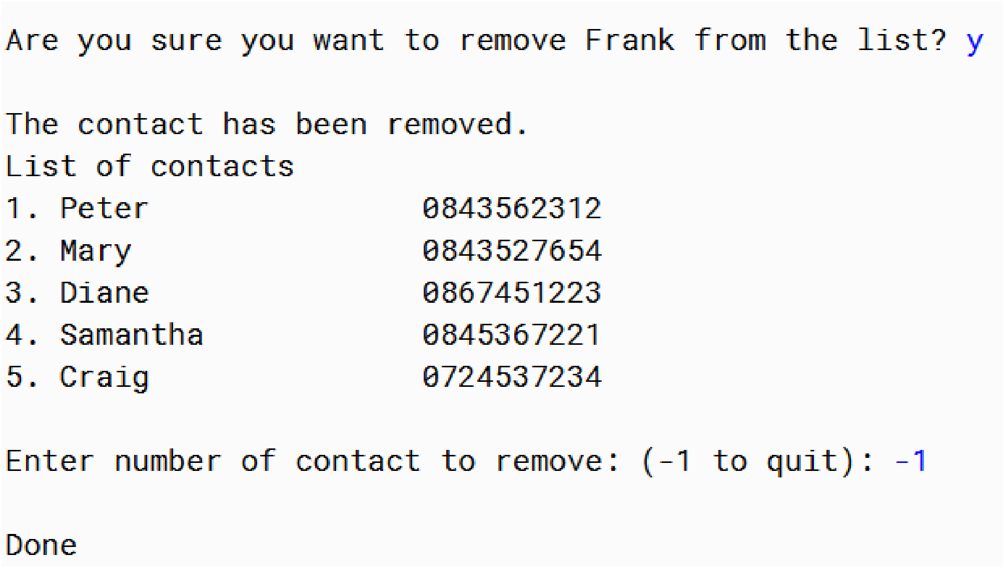
Example run 1:



```
List of contacts
1. Peter                0843562312
2. John                 0724537998
3. Mary                 0843527654
4. Diane                0867451223
5. Samantha             0845367221
6. Craig                0724537234
7. Frank                0864352321

Enter number of contact to remove: (-1 to quit): 2

Are you sure you want to remove John from the list? y

The contact has been removed.
```

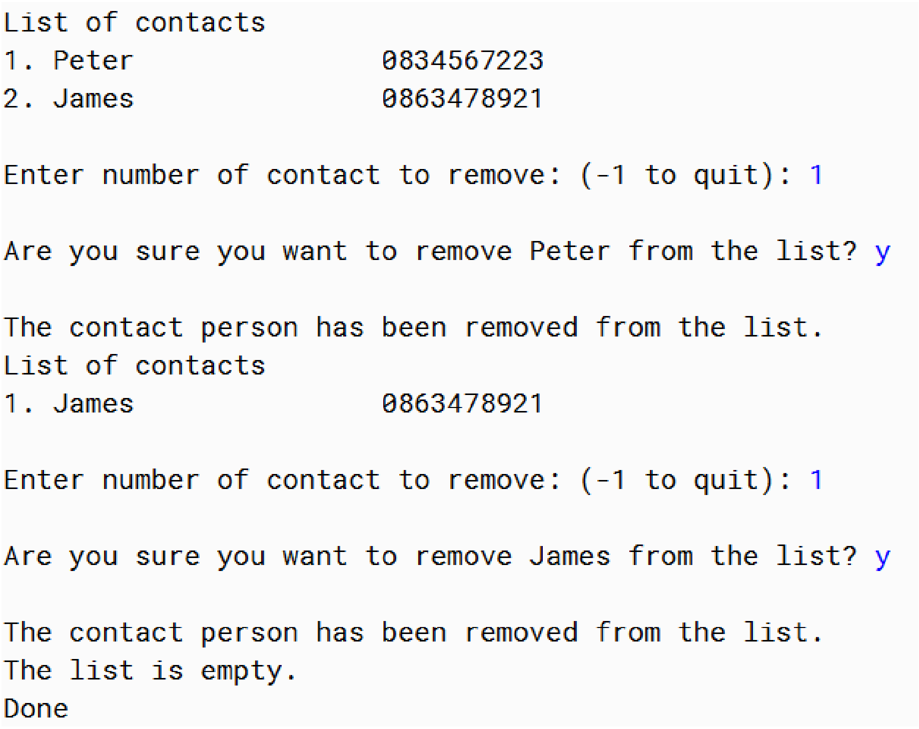


```
List of contacts
1. Peter               0843562312
2. Mary                0843527654
3. Diane               0867451223
4. Samantha            0845367221
5. Craig               0724537234
6. Frank               0864352321

Enter number of contact to remove: (-1 to quit): 6
```



```
Are you sure you want to remove Frank from the list? y

The contact has been removed.
List of contacts
1. Peter               0843562312
2. Mary                0843527654
3. Diane               0867451223
4. Samantha            0845367221
5. Craig               0724537234

Enter number of contact to remove: (-1 to quit): -1

Done
```

**Example run 2:**



```
List of contacts
1. Peter               0834567223
2. James               0863478921

Enter number of contact to remove: (-1 to quit): 1

Are you sure you want to remove Peter from the list? y

The contact person has been removed from the list.
List of contacts
1. James               0863478921

Enter number of contact to remove: (-1 to quit): 1

Are you sure you want to remove James from the list? y

The contact person has been removed from the list.
The list is empty.
Done
```