```java
public class Contact {

	private String name, number;

	public Contact(String name, String number) {
		super();
		this.name = name;
		this.number = number;
	}

	public String getName() {
		return name;
	}

	public void setName(String name) {
		this.name = name;
	}

	public String getNumber() {
		return number;
	}

	public void setNumber(String number) {
		this.number = number;
	}

	@Override
	public boolean equals(Object obj) {
		// TODO Auto-generated method stub
		if(obj == null ||
			this.getClass()!= obj.getClass())
			return false;
		Contact temp = (Contact) obj;
```

```
            if(this.name.equals(temp.getName()) &&
                this.number.equals(temp.getNumber()))
                return true;
            return false;
        }

        @Override
        public String toString() {
            return "Contact [name=" + name + ", number=" +
number + "]";
        }
}
```

```java
import java.util.Arrays;
public class Group {

    private String name;
    private Contact contacts[];
    private int nbC;

    public Group(String name, int size){
        this.name = name;
        contacts = new Contact[size];
        nbC = 0;
    }

    public Contact[] getContacts() {
        return contacts;
    }

    public int getNbC() {
        return nbC;
    }
    public boolean isFull(){
        return nbC >= contacts.length;
    }
    public boolean isEmpty(){
        return nbC == 0;
    }

    public int getIndex(Contact C){
        if(isEmpty()) return -1;
        for(int i = 0; i < nbC; i++)
            if(contacts[i].equals(C))
                return i;
        return -1;
    }
```

```java
public boolean addContact(Contact C){
    if(isFull())
        return false;
    contacts[nbC++] = C;
    return true;
}

public boolean removeContact(Contact C){
    int index = getIndex(C);
    if(index == -1) return false;
    contacts[index] = contacts[--nbC];
    return true;
}

public void printContactAt(int position){
    if(position < 0 || position >= nbC){
        System.out.println("Error");
    }
    else{
        System.out.println(contacts[position]);
    }
}
public Contact[] concat(Group g){
    int size = this.nbC + g.getNbC();
    Contact tempArr[] = new Contact[size];
    int counter = 0;
    for(int i = 0; i < nbC; i++){
        tempArr[counter++] = contacts[i];
    }
    for(int i = 0; i < g.getNbC(); i++)
        tempArr[counter++] = g.getContacts()[i];
    return tempArr;
}
```

```
    @Override
    public String toString() {
        return "Group [name=" + name + ", contacts="
                + Arrays.toString(contacts) + "]";
    }
}

public class testGroup {

    public static void main(String[] args) {
        // TODO Auto-generated method stub

        Group g1 = new Group("Family", 3);
        Group g2 = new Group("Friends", 4);
        g1.addContact(new Contact("Father","0555555"));
        g1.addContact(new Contact("Mother","0555556"));
        g1.addContact(new Contact("Brother",
                                  "0555557"));
        System.out.println(g1);
        g2.addContact(new Contact("Mohammed",
                                  "055554"));
        g2.addContact(new Contact("Ali", "055559"));
        g2.addContact(new Contact("Omar", "055551"));
        g2.addContact(new Contact("Khalid", "055552"));
        System.out.println(g2);
        Contact temp[] = g1.concat(g2);
        for(Contact C: temp){
            System.out.println(C);
        }
    }
}
```