1. Create a Project to calculate the binary number correspondent to a number asked to the user. These are the steps:
    a. Ask the user a decimal number.
    b. Create an ArrayList of integer numbers.
    c. In a loop, we are going to divide the decimal number between 2 and storing the remainder in the List. When the quotient is zero, we stop dividing.
    d. We show the elements from the List in inverse order and we will have the binary number.

   For example:
   Input a number: 46
   The binary number is: 101110

   Here you have the internal proceeding, you don't have to show it on the screen:
   46/2=23 quotient 0 → add 0 to the list
   23/2=11 quotient 1 → add 1 to the list
   11/2=5 quotient 1 → add 1 to the list
   5/2 =2 quotient 1 → add 1 to the list
   2/2=1 quotient 0 → add 0 to the list
   1/2=0 quotient 1 → add 1 to the list

2. Create a program that asks integer numbers to the user until they press enter.
    a. These numbers have to be added to an ArrayList.
    b. Then show them in inverse order.
    c. And then ask the user a number and we have to show on the screen the position of this number in the ArrayList.

3. Create a program to manage a phone list using a Map. The key will be a string with the name + surname and the value will be the phone number. You have to show this menu on the screen:
    a. **Add** (add a new number to the list)
    b. **Delete** (delete a number. You will have to show all and ask which one we want to delete)
    c. Show all
    d. **Modify** (You will have to show all and ask which one we want to modify the number. Remember that we only can modify the value, never the key)
    e. Exit

4. Create a Klingon dictionary using a Map. Here you have a dictionary of Klingon: http://es.startrek.wikia.com/wiki/Diccionario_Klingon

You can add some words and expression from this web and add them to your Map. Then ask the user a text in English and show the translation in Klingon.

5. Create a project called SchoolManagement with:
    a. A package called “model” with these classes:
    b. A class called **Person** with these attributes:
        i. name
        ii. dni
        iii. address
        iv. age
    c. A class called **Student** that inherits from Person with this new attribute:
        i. class
    d. A class called **Teacher** that inherits from Person with this new attribute:
        i. An ArrayList of classes (string)
    e. A class called **School** with this attribute:
        i. An ArrayList of Person objects.
        ii. A Map called Excursions: key→ name of the excursion, value → an ArrayList students.
    f. A package called “main” with the main program. In this program we will have a menu with these options:
        i. Add a new Person to School (you have to ask which type of person and ask the appropriate information).
        ii. Show all the students (use of instanceof)
        iii. Show all the teachers (use of instanceof)
        iv. Show the students of a class
        v. Create an excursion of a class. (You will add a new excursion to the Map of excursions. You have to ask the class that is going to go out and then show all the students from this class. Then you have to ask the students that are going to go on the trip and these students will be added to the List corresponding to this excursion).
        vi. Show the students that are going to go to one excursion ordered by name.

Excursion Map Example:

Key	Value
Madrid	ArrayList : 1.- Fernández Ramiro, Juan 2.- Romero Ramírez, Ana 3.- …
Granada	ArrayList: 1.- Abad Andreu, Anabel 2.- …
…	