Phonebook

You just moved to a new town, and you'd like to get new friends as soon as possible. You've decided to create a phone book application, or at least some parts of it first.

While you were experimenting with a friendly user interface, you attended a cool office party where you collected a few phone numbers. Later on you realized that many numbers are incorrect (maybe the party was too good), so you decided to write a program that selects the good ones, and also checks for their area codes.

What are you going to learn?

- Get user input.
- Concatenate strings.
- Write strings to the console.
- Use lists.
- Use conditional statements.
- Count items.

Tasks

Hello there

Ask for the user's name in `greetings.py`, and print out a friendly, personalized message.

1. The program asks for the user's name.
2. After the input, the program greets the user by the given name like this: `Hello XY, I am your phone book.`

CRITERIA ⌃

How old are you?

Still in in `greetings.py`, ask for the user's age, and print a polite message based on the answer. Only integers should be accepted.

1. If the input is not an integer, the following message is displayed: `That doesn't seem to be an integer.`
2. If the user is 18 or younger, the following message is displayed: `You are so young! Life is ahead of you!`
3. If the user is older than 18, but younger than 40, the following message is displayed: `That's a nice age!`
4. If the user is at least 40, the program prints out the following message: `You must be very wise!`

CRITERIA ⌃

Check the length

You found a few numbers in your phonebook that are shorter than the others so they are possibly wrong numbers. A properly formatted number should be 10 characters long (including dashes). Your task is to separate the good ones from the wrong ones based on the length, and print the two lists after each other. Formatting and dashes do not matter here.

1. The program selects the too short and too long numbers.
2. The program prints out the good phone numbers first, then the list of the wrong ones.

CRITERIA ⌃

Area codes

Print out the list of area codes and the list of phone numbers separately.

1. The area codes are printed as a separate list.
2. The phone numbers are printed without the area codes separately.
3. Both lists contain pieces of completely valid phone numbers.

CRITERIA ⌃

Unique area codes

Collect the unique area codes from the list.

1. The program prints out the list of all area codes (without duplicates).
2. The list contains area codes of completely valid phone numbers.

CRITERIA ⌃

Count area codes

Count how many (valid) phone numbers belong to the given area codes.

1. The program counts (without hardcoding the results) the phone numbers belonging to the three given area codes (98, 34, 72) and prints the results.
2. The counter only counts valid phone numbers.

CRITERIA ⌃

Valid pattern

You received another list of phone numbers (see phonebook.py) but most of them look ugly. A valid number should look like `nn-nnn-nnn`. Your task is to check every phone number in the list and filter those which are not valid.

1. The program selects and indicates the wrong numbers not following the pattern.
2. The program prints out the good phone numbers first, then the list of the wrong ones.
3. The program might use regular expression pattern matching.

CRITERIA ⌃

Hints

- Please use the given list of phone numbers, and do not modify it.
- The area number is the first two digits of the (valid) phone number.
- For number validation you can use the `isnumeric()` method on strings.
- You should use f-strings when printing, it makes your life easier.
- You can either remove the invalid numbers from the list, or just create a new list for the valid ones.
- The last task requires string pattern checking. For such tasks programmers usually use so-called "regular expressions" (a.k.a. regex)