

Phone Number Cleaner

Introduction

You work in a company with a huge number of phone numbers. Unfortunately, the data is a big mess. They have been entered in various different forms (e.g. +43722912345 or 0660-75099345 or 0043/1/034508). It is your job to implement a command-line helper tool that analyzes the phone numbers.

Basic Requirements

- Implement a command-line application in C#.
- The application receives a phone number as the command-line argument. If the program is called without any argument, it must print out an error message (*Missing phone number, please try again*).
- Any slashes (/) or dashes (–) that might appear in phone numbers must be ignored.
- Your program must check phone numbers based on the following rule set:
 - A phone number might be an emergency number. Valid emergency numbers are 112, 122, 133, and 144.
 - If the phone number is not an emergency number, ...
 - * ...it must start with +43 or 0043 or 0 and
 - * ...all remaining phone number characters must be digits and
 - * ...the length of the phone number must be at least 6 digits.
- If the phone number is an emergency number, your program must print out *Emergency number: {phone number}*.
- If the phone number is invalid, your program must print out *Invalid phone number: {phone number}*
- If the phone number is valid and not an emergency number, your program must print out the phone number with leading +43. Here is some test data illustrating how your program must convert phone numbers:

Input	Printed phone number
+4373212345	+4373212345

Input	Printed phone number
004373212345	+4373212345
073212345	+4373212345

- You must implement and use a method `IsValidPhoneNumber`. This method receives the phone number from the command line argument as the input parameter and returns true if the number is valid according to the rules defined above, otherwise false.
- You must implement and use a method `CleanupPhoneNumber`. This method receives the phone number from the command line argument and returns the cleaned version as shown in the table above. If you want, you can use a `ref` parameter instead of returning the cleaned phone number.

Advanced Requirements

- Your application must be able to process **one or many phone numbers** in the command-line. If the program is called without any argument, it must print out an error message (*Missing phone number(s), please try again*).
- In addition to the cleaned phone number, your program must separately print the area code and the phone number without area code. Here are the rules to extract area codes:
 - Vienna has the single-digit area code *1*.
 - The following area codes have three digits: *316, 512, 463, 732, 662, 660, 699, 664, 676, 650, 680, 681, 665, 688, 677, 670, 690*
 - All other area codes have four digits.
 - Emergency numbers do not have an area code.
- You must implement and use a method `SplitPhoneNumber`. This method receives the phone number from the command line argument and returns area code and phone number without area code. Use `out` parameter for returning both values.

Here is some test data with input and corresponding output:

Input	Expected output
112	Emergency number: 112
+4312abc	Invalid phone number: +4312abc

Input	Expected output
0043-1-9876543	+4319876543, Area Code 1, Phone Number 9876543
+43/732/12345	+4373212345, Area Code 732, Phone Number 12345
07229234567	+437229234567, Area Code 7229, Phone Number 234567
0123	Invalid phone number: 0123
01234567	+431234567, Area Code 1, Phone Number 234567