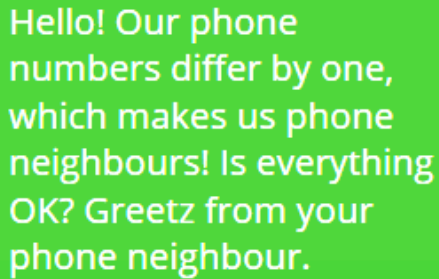


Phone neighbours

Better a good phone neighbour than a distant friend. With that idea in mind, the editorial officers of the digital experiment [Sambal](#) attempted to spark some new friendships around Flanders in December 2014. To test the new concept, each Sambal team member sent the following text message to his or her phone neighbours, hoping for a quick response.



Hello! Our phone numbers differ by one, which makes us phone neighbours! Is everything OK? Greetz from your phone neighbour.

Original text message sent from each Sambal team member to his or her phone neighbours, hoping for a quick response.

But what exactly are your **phone neighbours**? Well, simple enough, they are the people who have almost the same phone number as yourself, but then one number higher or lower. For example, if your phone number is 0472/91.39.17, then your **downstairs neighbour** has phone number 0472/91.39.16 and your **upstairs neighbour** has phone number 0472/91.39.18.

Getting to know your phone neighbours is a two-step process. At first, off course, you send them a text message. If you receive an answer, you take a screenshot of the conversation that follows. The best and funniest screenshots are collected on a temporary [Facebook page](#).

Assignment

In this assignment, we represent a phone number as a string that contains both digits and other characters. Each phone number contains at least one digit. The other characters are used to format the phone number in some way or another. As an example, the phone number 0472/91.39.17 is formatted using one extra slash and three extra dots. The goal of this assignment is to determine the phone neighbours of a given phone number, and express them in the same format as the original phone number. This is done in the following way:

- Write a function `digits` that takes a phone number as its argument. The function must return a string that only contains the digits of the given phone number, in the order in which they appear in the given phone number.
- Write a function `replace` that takes two arguments: a phone number and a positive integer (an `int`). The function must replace each of the digits of the given phone number by the digits of the given integer and return the result. In case the given phone number contains more digits than the given integer, the extra digits in front of the phone number must be replaced by zeros. In case the given phone number contains less digits than the given integer, the excess digits at the end of the integer should not be used in the substitution.
- Use the functions `digits` and `replace` to write a function `upstairsNeighbour` that takes a phone number as its argument. The function must return the upstairs neighbour of the given phone number.

- Use the functions `digits` and `replace` to write a function `downstairsNeighbour` that takes a phone number as its argument. The function must return the downstairs neighbour of the given phone number. The function may assume that not all digits of the given phone number are zeros.

Example

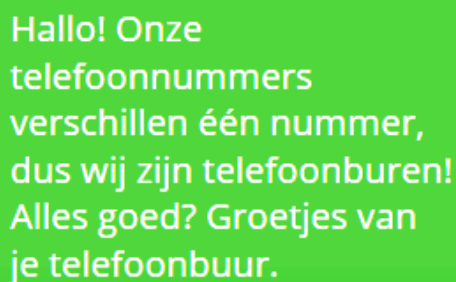
```
>>> digits('0472/91.39.17')
'0472913917'
>>> digits('++32 (0)9 264 4779')
'32092644779'

>>> replace('0472/91.39.17', 1234567890)
'1234/56.78.90'
>>> replace('++32 (0)9 264 4779', 123456789)
'++00 (1)2 345 6789'

>>> upstairsNeighbour('0472/91.39.17')
'0472/91.39.18'
>>> upstairsNeighbour('++32 (0)9 264 4779')
'++32 (0)9 264 4780'

>>> downstairsNeighbour('0472/91.39.17')
'0472/91.39.16'
>>> downstairsNeighbour('++32 (0)9 264 4779')
'++32 (0)9 264 4778'
```

Beter een goede telefoonbuur dan een verre vriend. Met dat idee ging de redactie van de experimentele nieuwssite [Sambal](#) van de VRT in december 2014 aan de slag om mensen in Vlaanderen warm te maken om nieuwe vriendschappen aan te gaan. Om het concept uit te testen, stuurde iedereen op de redactie van Sambal alvast onderstaande sms naar zijn of haar telefoonburen, hopen op een antwoord.



Hallo! Onze
telefoonnummers
verschillen één nummer,
dus wij zijn telefoonburen!
Alles goed? Groetjes van
je telefoonbuur.

De oorspronkelijke sms die iedereen op de redactie van Sambal stuurde naar zijn of haar telefoonburen, hopen op een antwoord.

Maar wat zijn nu eigenlijk je **telefoonburen**? Wel, heel eenvoudig, het zijn de personen die bijna hetzelfde telefoonnummer hebben als jezelf, maar dan eentje hoger of lager. Is jouw telefoonnummer bijvoorbeeld 0472/91.39.17, dan heeft je **onderbuur** het telefoonnummer 0472/91.39.16 en je **bovenbuur** het telefoonnummer 0472/91.39.18.

Het leren kennen van je telefoonburen gaat in twee stappen. De eerste stap is natuurlijk een bericht sturen. Als er antwoord komt, dan wordt er van de conversatie een screenshot genomen. De beste en leukste screenshots werden verzameld op een tijdelijke [Facebookpagina](#).

Opgave

In deze opgave stellen we een telefoonnummer voor als een string die zowel cijfers als andere karakters kan bevatten. Een telefoonnummer bevat altijd minstens één cijfer. De andere karakters zorgen voor een bepaalde opmaak van het telefoonnummer. Zo hebben we bij de voorstelling van het telefoonnummer 0472/91.39.17 voor de opmaak bijvoorbeeld gebruik gemaakt van een extra slash en drie extra punten. Het doel van deze opgave is om de telefoonburen te bepalen van een gegeven telefoonnummer, met behoud van de opmaak van het telefoonnummer. Hiervoor gaan we als volgt te werk:

- Schrijf een functie `cijfers` waaraan een telefoonnummer moet doorgegeven worden. De functie moet een string teruggeven die enkel bestaat uit de cijfers van het gegeven telefoonnummer, in de volgorde waarin ze in het telefoonnummer voorkomen.
- Schrijf een functie `vervangen` waaraan twee argumenten moeten doorgegeven worden: een telefoonnummer en een natuurlijk getal (een `int`). De functie moet de opeenvolgende cijfers van het gegeven telefoonnummer vervangen door de opeenvolgende cijfers van het gegeven getal, en het resultaat daarvan teruggeven. Indien het gegeven telefoonnummer meer cijfers bevat dan het gegeven getal, dan moeten de extra cijfers vooraan vervangen worden door nullen. Indien het gegeven telefoonnummer minder cijfers bevat dan het gegeven getal, dan moeten de overtollige cijfers op het einde van het getal niet gebruikt worden bij de vervanging.
- Gebruik de functies `cijfers` en `vervangen` om een functie `bovenbuur` te schrijven waaraan een telefoonnummer moet doorgegeven worden. De functie moet de bovenbuur van het gegeven telefoonnummer teruggeven.
- Gebruik de functies `cijfers` en `vervangen` om een functie `onderbuur` te schrijven waaraan een telefoonnummer moet doorgegeven worden. De functie moet de onderbuur van het gegeven telefoonnummer teruggeven. Hierbij mag de functie ervan uitgaan dat de cijfers van het gegeven telefoonnummer niet allemaal nullen zijn.

Voorbeeld

```
>>> cijfers('0472/91.39.17')
```

```
'0472913917'
```

```
>>> cijfers('++32 (0)9 264 4779')
```

```
'32092644779'
```

```
>>> vervangen('0472/91.39.17', 1234567890)
```

```
'1234/56.78.90'
```

```
>>> vervangen('++32 (0)9 264 4779', 123456789)
```

```
'++00 (1)2 345 6789'
```

```
>>> bovenbuur('0472/91.39.17')
```

```
'0472/91.39.18'
```

```
>>> bovenbuur('++32 (0)9 264 4779')
```

```
'++32 (0)9 264 4780'
```

```
>>> onderbuur('0472/91.39.17')
```

```
'0472/91.39.16'
```

```
>>> onderbuur('++32 (0)9 264 4779')
```

```
'++32 (0)9 264 4778'
```