

# Call forwarding

Thanks to computer technology the functionality of phone systems has been greatly enhanced in the last ten years. We now have automated menus, sophisticated answering machines, conference call capabilities, group addressing and so on. A common feature of most current phone systems is the ability to set call forwarding. If somebody goes on vacation, he sets things up so that all incoming calls are forwarded to a colleague. Your task is to write a program that can be used to manage call forwarding at Ghent University.

All phones at Ghent University have four digit extensions. Employees can set call forwarding by entering the appropriate information through their telephone interface. If an employee is going to be away he enters the following information: his extension, the extension his calls should be forwarded to, the time he will be leaving, and how long he will be away. This information is subject to the following constraints:

- all phone extensions consist of four digits, where leading zeros are allowed
- the phone extension 9999 is reserved and has special meaning for the call forwarding system (see below)
- times are recorded in increments of one hour and are based on a clock that begins at 0 at midnight every New Year's Eve; therefore, when describing the time they are leaving, employees always use an integer between 0 and 8784 ( $= 366 \times 24$ ); the call forwarding system is completely reset at the beginning of a year
- a call forward set at time  $t$  for a duration of  $d$  will be in effect from time  $t$  to time  $t + d$  inclusive

The call forwarding system may assume that employees enter "correct" information, in the sense that they follow the formatting rules and they do not enter a request such that the duration of the request would go past the end of the year. The system must explicitly check if the same employee does not enter multiple requests that overlap in time. But even though employees enter correct, clear, non-overlapping information from their own point of view, inconsistencies can still occur in the call forwarding system if requests have been made in such a way as to forward a call back to the original target of the call. For example, if Bob forwards his calls to Sue, and Sue forwards her calls to Joe, and Joe forwards his calls to Bob. When somebody calls any of these three people their calls would be forwarded forever. To prevent this situation the call forwarding system uses the *dead end* phone extension 9999. Any calls made to an extension involved in such a degenerate situation will be forwarded to the special phone extension 9999.

## Assignment

Define a class `CallForwarding` that can be used to operate a phone dispatching center that supports call forwarding. The functionality of the dispatching center must reflect the one described above. Objects of the class `CallForwarding` must therefore at least support the following methods:

- A method `setForward` that can be used to enter a new request to the dispatching center. The method takes four arguments: *i*) the phone extension (`string`) of the employee that enters the request, *ii*) the phone extension (`string`) to which the call must be forwarded, *iii*) the time  $t$  (`int`) at which the forward will become in effect, and *iv*) the duration  $d$  (`int`) of the forward. If the employee enters a request that overlaps in time with one of his previous requests, the

method must throw an `AssertionError` with message `invalid forward`.

- A method `setForwards` that takes the name of a text file as an argument. Each line of the text file contains four integers (separated by spaces), where each number consists of four digits, where each integer consists of four digits. These integers represent the information for entering a call forward request: *source target time duration*. For each request, the method must set up a call forwarding from the *source* to the *target* starting at the *time* for a length of *duration*.
- A method `call` that can be used to process incoming calls. This method takes two arguments that represent an incoming call to a given phone extension (string, first argument) at a given point in time (int, second argument). The method must return a string as its result, representing the phone extension to which the incoming call will be finally forwarded. Forwarding should be done according to the call forwarding requests that have been entered into the dispatching center previous to the method call.

## Example

In the following interactive session we assume the current directory to contain the text file [settings.txt](#).

```
>>> dispatch = CallForwarding()
>>> dispatch.call('4444', 150)
'4444'
>>> dispatch.setForward('4444', '6666', 100, 200)
>>> dispatch.setForward('5555', '7777', 600, 200)
>>> dispatch.call('5555', 700)
'7777'
>>> dispatch.setForward('5555', '8888', 750, 100)
Traceback (most recent call last):
AssertionError: invalid forward
```

```
>>> dispatch2 = CallForwarding()
>>> dispatch2.setForwards('settings.txt')
>>> dispatch2.call('1111', 50)
'1111'
>>> dispatch2.call('1111', 150)
'2222'
>>> dispatch2.call('1111', 200)
'3333'
>>> dispatch2.call('2222', 225)
'3333'
>>> dispatch2.call('1111', 270)
'9999'
>>> dispatch2.call('1111', 320)
'4444'
>>> dispatch2.call('3333', 320)
'4444'
>>> dispatch2.call('3000', 900)
'3000'
>>> dispatch2.call('3333', 1250)
'1111'
>>> dispatch2.call('7777', 1250)
'9999'
```

Door toedoen van computertechnologie is de functionaliteit van telefoniesystemen de laatste tien jaar sterk toegenomen. We beschikken nu over geautomatiseerde menu's, gesofisticeerde

antwoordapparaten, *conference call* mogelijkheden, groepsadressering, en dergelijke meer. Een functionaliteit die de meeste telefoniesystemen aanbieden, is om oproepen door te schakelen. Als iemand op vakantie gaat, dan kan hij het systeem instellen zodat binnenkomende oproepen worden doorgeschakeld naar een collega. Voor deze opgave moet je een programma schrijven dat kan gebruikt worden om doorgeschakelde oproepen te beheren voor de Universiteit Gent.

Alle telefoonnummers van de Universiteit Gent bestaan uit vier cijfers. Werknemers kunnen oproepen doorschakelen door bepaalde informatie in te voeren via hun telefoontoestel. Als een werknemer afwezig zal zijn, dan voert hij de volgende informatie in: zijn eigen telefoonnummer, het telefoonnummer waarnaar moet doorgeschakeld worden, het tijdstip vanaf wanneer hij afwezig zal zijn en hoe lang hij afwezig zal zijn. Deze informatie is onderhevig aan de volgende voorwaarden:

- telefoonnummers bestaan altijd uit vier cijfers, waarbij vooraan ook nullen kunnen staan
- het telefoonnummer 9999 is voorbehouden, en heeft een speciale betekenis voor het doorschakelsysteem (zie verder)
- tijdstippen worden voorgesteld als incrementen van één uur en zijn gebaseerd op een klok die telkens bij het begin van het jaar begint te lopen vanaf 0; het tijdstip vanaf wanneer een werknemer afwezig zal zijn, wordt dus altijd weergegeven als een geheel getal tussen 0 en 8784 ( $= 366 \times 24$ ); het doorschakelsysteem wordt telkens herstart bij het begin van een nieuw jaar
- een doorschakelopdracht die wordt ingesteld vanaf tijdstip  $t$  voor een duurtijd  $d$  zal actief zijn vanaf tijdstip  $t$  tot en met tijdstip  $t + d$

De telefooncentrale mag ervan uitgaan dat werknemers "correcte" informatie doorgeven aan het doorschakelsysteem. Ze geven de informatie in volgens het juiste formaat. Ze geven geen doorschakelopdrachten die het einde van het jaar overschrijden. De centrale moet wel controleren of eenzelfde werknemer nooit meerdere doorschakelopdrachten ingeeft waarvan de periodes overlappen in de tijd. Maar zelfs als de werknemers informatie doorgeven die vanuit hun eigen standpunt correct is, kunnen er nog inconsistenties optreden waarbij er oneindig wordt doorgeschakeld. Stel bijvoorbeeld dat Nico zijn oproepen doorschakelt naar Veerle, Veerle schakelt haar oproepen door naar Annick, en Annick schakelt haar oproepen door naar Nico. Als één van deze drie werknemers wordt opgebeld, dan zal de oproep tot het einde der tijden doorgeschakeld worden. Om dergelijke situaties op te vangen, gebruikt het doorschakelsysteem het *doodlopend* telefoonnummer 9999. Elke binnenkomende oproep die aanleiding geeft tot een oneindige lus wordt daarbij doorgeschakeld naar het speciale telefoonnummer 9999.

## Opgave

Definieer een klasse Telefooncentrale waarmee het doorschakelsysteem van een telefooncentrale kan aangestuurd worden. De functionaliteit van het doorschakelsysteem moet hierbij voldoen aan bovenstaande omschrijving. De objecten van de klasse Telefooncentrale moeten daarvoor minstens over de volgende methoden beschikken:

- Een methode doorschakelen waarmee een doorschakelopdracht kan doorgeven worden aan de telefooncentrale. Aan deze methode moeten vier argumenten doorgegeven worden: *i*) het telefoonnummer (string) van de werknemer die de doorschakelopdracht ingeeft, *ii*) het telefoonnummer (string) waarnaar moet doorgeschakeld worden, *iii*) het tijdstip  $t$  (int) vanaf wanneer de doorschakeling moet ingaan, en *iv*) de duurtijd  $d$  (int) van doorschakeling.

Indien de werknemer die de doorschakelopdracht ingeeft eerder al een doorschakelopdracht heeft ingegeven die in de tijd overlapt met de nieuwe doorschakelopdracht, dan moet de methode een `AssertionError` opwerpen met als boodschap ongeldige opdracht.

- Een methode `configureer` waaraan de naam van een tekstbestand moet doorgegeven worden. Elke regel van het tekstbestand bevat vier getallen (gescheiden door spaties) bestaande uit vier cijfers. Deze getallen bevatten de informatie voor het instellen van een doorschakelopdracht: *doel bestemming start duur*. De methode moet deze doorschakelopdrachten doorgeven aan de telefooncentrale, waarbij het doorschakelsysteem zo ingesteld wordt dat een oproep naar werknemer met telefoonnummer *doel* moet doorgeschakeld worden naar telefoonnummer *bestemming* vanaf tijdstip *start* voor een zekere periode van *duur* uur.
- Een methode `oproep` waarmee een binnenkomende oproep kan verwerkt worden. Aan deze methode moeten twee argumenten doorgegeven worden die aangeven dat er op een gegeven tijdstip (int, tweede argument) een oproep binnenkomt naar een gegeven telefoonnummer (string, eerste argument). De methode moet een string als resultaat teruggeven, die het telefoonnummer aangeeft waarnaar de binnenkomende oproep zal doorgeschakeld worden volgens de doorschakelopdrachten die tot hiertoe aan het systeem doorgegeven werden.

## Voorbeeld

Bij onderstaande voorbeeldsessie gaan we ervan uit dat het tekstbestand [instellingen.txt](#) zich in de huidige directory bevindt.

```
>>> centrale = Telefooncentrale()
>>> centrale.oproep('4444', 150)
'4444'
>>> centrale.doorschakelen('4444', '6666', 100, 200)
>>> centrale.doorschakelen('5555', '7777', 600, 200)
>>> centrale.oproep('5555', 700)
'7777'
>>> centrale.doorschakelen('5555', '8888', 750, 100)
Traceback (most recent call last):
AssertionError: ongeldige opdracht
```

```
>>> centrale2 = Telefooncentrale()
>>> centrale2.configureer('instellingen.txt')
>>> centrale2.oproep('1111', 50)
'1111'
>>> centrale2.oproep('1111', 150)
'2222'
>>> centrale2.oproep('1111', 200)
'3333'
>>> centrale2.oproep('2222', 225)
'3333'
>>> centrale2.oproep('1111', 270)
'9999'
>>> centrale2.oproep('1111', 320)
'4444'
>>> centrale2.oproep('3333', 320)
'4444'
>>> centrale2.oproep('3000', 900)
'3000'
```

```
>>> centrale2.oproep('3333', 1250)
'1111'
>>> centrale2.oproep('7777', 1250)
'9999'
```