# IF100
# Nested Loops - Practice #1

## Introduction

The aim of this example is to practice sequences (i.e. list, string, range) and loops (for and while statements, and nested loops). The use of sequences and loops is due to the nature of the problem; that is, you cannot finish this practice assignment without using sequences and loops.

## Description

In this exercise, you will implement the D'Hondt method, the method used for calculating how many seats each party will get in the parliament after an election. You will have a chance to work on a real dataset which is the result of the general election of 2002 in Turkey.

Proportional representation systems aim to allocate seats to the parties approximately in proportion to the number of votes received. For example, if a party wins one-third of the votes then it should gain about one-third of the seats. In general, exact proportionality is not possible because these divisions produce fractional numbers of seats. As a result, several methods, of which the D'Hondt method is one, have been devised which ensures that the parties' seat allocations, which are of course whole numbers, are as proportional as possible.

Legislatures using this system include those of Albania, Argentina, Armenia, Austria, Belgium, Brazil, Bulgaria, Cambodia, Cape Verde, Chile, Colombia, Croatia, the Czech Republic, Denmark, the Dominican Republic, East Timor, Ecuador, Estonia, Fiji, Finland, Guatemala, Hungary, Iceland, Israel, Japan, Kosovo, Luxembourg, Macedonia, Moldova, Montenegro, Netherlands, Northern Ireland, Paraguay, Peru, Poland, Portugal, Romania, Scotland, Serbia, Slovenia, Spain, Turkey, Uruguay, and Wales.

Typically, the country is divided into *electoral districts* (also called as *voting districts*, or simply *districts*). In Turkey, each city is considered to be a separate electoral district. For some elections, big cities (like Istanbul, Ankara, etc.) are divided into smaller electoral districts (like Istanbul-1, Istanbul-2, etc). However, in this homework, we will consider each city as an electoral district. Each electoral district is given a certain number of seats, which is the total number of members of parliament to be elected from that district. For example, Istanbul sends 70 members, whereas Bolu sends 3 members to the parliament, which has 550 seats in total.

After the votes for the parties are counted within an electoral district, the D’Hondt method is used to calculate the number of seats each party will get in that district. The D’Hondt method is an iterative process, where each party is given a number

$$N = \frac{V}{(s+1)}$$

where

- ***V*** is the total number of votes that party received, and
- ***s*** is the number of seats that party has been allocated so far (***s*** is initially 0 for all parties).

At each stage, the party with the highest ***N*** value is given a seat, and that party’s ***s*** value goes up by one. This iterative process continues until all seats in the electoral district is allocated.

Let’s give an example of how the D’Hondt algorithm works. Assume that a given electoral district has 5 seats to be allocated; and there are 3 parties, called as Party A, Party B, and Party C. Let’s say in this district Party A, B, C got 100, 80 and 35 votes respectively.

| STAGE | Seats taken by Party A so far (s for A) | Seats taken by Party B so far (s for B) | Seats taken by Party C so far (s for C) | N for Party A<br>(V=100) | N for Party B<br>(V=80) | N for Party C<br>(V=35) |
|---|---|---|---|---|---|---|
| 1 | 0 | 0 | 0 | **100** | 80 | 35 |
| 2 | 1 | 0 | 0 | 50 | **80** | 35 |
| 3 | 1 | 1 | 0 | **50** | 40 | 35 |
| 4 | 2 | 1 | 0 | 33.33 | **40** | 35 |
| 5 | 2 | 2 | 0 | 33.33 | 26.66 | **35** |
| RESULT | 2 | 2 | 1 | 33.33 | 26.66 | 17.5 |

Bold values in each row of the table indicates the maximum **N** value in that stage. For the example above, the D’Hondt method will allocate 2 seats to Party A, 2 seats to Party B, and 1 seat to Party C.

## Prepared Dataset

For this example, we provide you with a real dataset which was given in the shared colab file, and this file contains the election results of 2002 in Turkey (we have omitted independent candidates for the simplicity of the homework). We have some variables that contain information regarding the elections. We will now explain these variables.

```
partyNames = ["Adalet ve Kalkınma Partisi", "Cumhuriyet Halk Partisi",
"Doğru Yol Partisi", "Milliyetçi Hareket Partisi", "Genç Parti"]
```

*partyNames* is actually a list of strings, and it contains the names of the political parties.

```
countrywidePercentages = [34.28, 19.39, 9.54, 8.36, 7.25]
```

*countrywidePercentages* is another list which contains float values. These float values are percentage of votes that the parties got across the country. Any index of the *countrywidePercentages* represents the percentage of the corresponding party that is located in the same index of *partyNames* list. For instance, *"Adalet ve Kalkınma Partisi"* has 34.28 percent of all the votes, *"Cumhuriyet Halk Partisi"* has 19.39 percent of all the votes across the country. Hence when you look at the data, you will realize that the length of *partyNames* and *countrywidePercentages* are the same.

We have a long list called *electoralDistricts* containing the names of the electoral districts in Turkey. For each electoral district, the deputy numbers (i.e. the number of seats) for that district is given at the same index of the list *districtDeputyNumbers*. For instance, *electoralDistricts[0]* is "Adana" and *districtDeputyNumbers[0] is* 14, which means that Adana has 14 seats. Similarly, *electoralDistricts[1]* is "Adıyaman" and *districtDeputyNumbers[1] is* 5, which means that Adıyaman has 5 seats. Again, when you look at the data, you will realize that the length of *electoralDistricts* and *districtDeputyNumbers* are the same.

We have one more variable which is *voteNumbersOfParties*. This variable is actually a list of lists; and every sublist represents the number of votes of each political party for the corresponding electoral district which is located at the same index of *electoralDistricts* list. Additionally, every integer value in these sublists indicates the corresponding vote number of the party that is located in the same index of *partyNames* list. For instance, very first sublist at the 0th index of *voteNumbersOfParties* is the following sublist:

```
[229729, 182436, 98440, 97700, 75464]
```

This sublist represents the number of votes of each political party for the electoral district located at the 0th index of *electoralDistricts*, which is "Adana". Additionally, 229729 is the number of votes of *"Adalet ve Kalkınma Partisi"* in "Adana", 182436 is the number of votes of *"Cumhuriyet Halk Partisi"* in "Adana", and so on. Similarly, the length of *voteNumbersOfParties* is equal to the length of *electoralDistricts* and *districtDeputyNumbers*.

### Inputs and Outputs

There will be only a single input of your program, the index of the district. Since we have 81 districts in our data, the input should be between 0 and 80 (both inclusive). Otherwise, you should give an error message and terminate the program. Additionally, you may assume that the input is composed of digits.

If the input entered is in the correct range, then you should print the name of the district and the number of deputies allocated for each party in that district on separate lines, after calculating the allocated deputy counts for each party by using the D'Hondt algorithm. Note that, while calculating the allocated deputy counts for each political party, you will assume that there is no threshold (barrage) value across the country. Thus, we will consider each vote of the political party even if they got a small number of votes in a district.

## Sample Runs

Below, we provide some sample runs of the program that you will develop. The *italic* and **bold** phrases are inputs taken from the user. You may not change any of the prompt sentences. Your program should be presented exactly like these sample runs.

Sample runs are not %100 comprehensive. You are required to read the whole documentation and decide on what other cases you might try your program with.

### Sample Run 1

```
Please enter an index between 0 and 80 (both are inclusive): 81
You entered an invalid input!!!
```

### Sample Run 2

```
Please enter an index between 0 and 80 (both are inclusive): 0
Results for Adana as follows:
Adalet ve Kalkınma Partisi: 5
Cumhuriyet Halk Partisi: 4
Doğru Yol Partisi: 2
Milliyetçi Hareket Partisi: 2
Genç Parti: 1
```

### Sample Run 3

```
Please enter an index between 0 and 80 (both are inclusive): 20
Results for Bursa as follows:
Adalet ve Kalkınma Partisi: 9
Cumhuriyet Halk Partisi: 3
Doğru Yol Partisi: 2
Milliyetçi Hareket Partisi: 1
Genç Parti: 1
```

### Sample Run 4

```
Please enter an index between 0 and 80 (both are inclusive): 1
Results for Adıyaman as follows:
Adalet ve Kalkınma Partisi: 4
Cumhuriyet Halk Partisi: 1
Doğru Yol Partisi: 0
Milliyetçi Hareket Partisi: 0
Genç Parti: 0
```

### Sample Run 5

```
Please enter an index between 0 and 80 (both are inclusive): 37
Results for Iğdır as follows:
Adalet ve Kalkınma Partisi: 0
Cumhuriyet Halk Partisi: 1
Doğru Yol Partisi: 0
Milliyetçi Hareket Partisi: 1
Genç Parti: 0
```