everis Python Exercise

Introduction

This exercise aims to evaluate the Python skills for the Python Developer role and it is designed to assess the technical proficiency and problem solving abilities of the candidate.

In this document you will find all the instructions to carry out the exercise and when concluded, the solution script needs to be committed to GitHub.

The candidate should be comfortable to contact us in case of any question.

The Use Case

You should receive a spreadsheet containing the needed data for this exercise.
In this spread sheet you will find some tables that will be used to compose the final result.

The final results must have this behaviour:

a) When run the function with any header of the main table:
get_trend('XPI')

b) The expected output needs to have all years for the selected trend as represented (illustration as the results needs to bring all years) below:

YEAR	XPI
1990	2.0
1991	1.9
1992	1.9
1993	1.8
1994	1.8
1995	1.7
1996	1.7
1997	1.6
1998	1.6
1999	1.6
2000	1.5
2001	1.5
2002	1.5
2003	1.4
2004	1.4
2005	1.3
2006	1.3
2007	1.3
2008	1.2
2009	1.2

Detailed Explanation

Please look at the spreadsheet and note that to achieve the results you will need to use the AUX, XPI, and WWW tables in order to apply all the necessary calculations as demonstrated.

Your code needs to cover all the calculations dynamically and you must use all the percentual indexes.

			3.0%		1.0%	2.0%		1.0%	2.0%
TRENDS									
	YEAR	FIXED	XPI	XPI +1.0%	XPI +2.0%	WWW	WWW +1.0%	WWW +2.0%
	1990	2.50	2.00	2.02	2.06	2.57	2.60	2.65
	1991	2.43	1.92	1.94	1.98	2.48	2.50	2.55
	1992	2.36	1.86	1.88	1.92	2.36	2.38	2.43
	1993	2.29	1.81	1.83	1.86	2.34	2.36	2.41
	1994	2.22	1.76	1.78	1.82	2.28	2.30	2.34
	1995	2.16	1.72	1.73	1.77	2.19	2.21	2.25
	1996	2.09	1.67	1.68	1.72	2.09	2.11	2.15
	1997	2.03	1.63	1.64	1.68	1.97	1.99	2.03
	1998	1.97	1.60	1.62	1.65	1.87	1.89	1.93
	1999	1.92	1.57	1.58	1.62	1.78	1.79	1.83
	2000	1.86	1.52	1.53	1.56	1.68	1.70	1.73
	2001	1.81	1.48	1.49	1.52	1.64	1.66	1.69
	2002	1.75	1.45	1.47	1.50	1.63	1.64	1.67
	2003	1.70	1.42	1.43	1.46	1.59	1.60	1.63
	2004	1.65	1.38	1.40	1.42	1.52	1.53	1.56
	2005	1.60	1.34	1.35	1.38	1.46	1.48	1.51
	2006	1.56	1.30	1.31	1.33	1.40	1.41	1.44
	2007	1.51	1.26	1.27	1.30	1.34	1.35	1.38
	2008	1.47	1.21	1.23	1.25	1.31	1.32	1.35
	2009	1.43	1.22	1.23	1.25	1.33	1.34	1.37
	2010	1.38	1.20	1.21	1.23	1.30	1.31	1.34

AUX

2020	258.81
2021	261.40
INFLATION	1.0%

XPI

Year	Value
1990	127.4
1990	128.0
1990	128.7
1990	128.9
1990	129.2
1990	129.9
1990	130.4
1990	131.6
1990	132.7
1990	133.5
1990	133.8
1990	133.8
1991	134.6
1991	134.8
1991	135.0
1991	135.2
1991	135.6
1991	136.0
1991	136.2
1991	136.6
1991	137.2
1991	137.4
1991	137.8
1991	137.9
1992	138.1

WWW

Year	Value
1985	16,822.51
1986	17,321.82
1987	18,426.51
1988	19,334.04
1989	20,099.55
1990	21,027.98
1991	21,811.60
1992	22,935.42
1993	23,132.67
1994	23,753.53
1995	24,705.66
1996	25,913.90
1997	27,426.00
1998	28,861.44
1999	30,469.84
2000	32,154.82
2001	32,921.92
2002	33,252.09
2003	34,064.95
2004	35,648.55
2005	36,952.94
2006	38,651.41
2007	40,405.48
2008	41,334.97
2009	40,711.61

everis
an NTT DATA Company

Assumptions and Considerations

Please ensure that you are covering all points listed below:

MUST TO HAVE:

- ❑ All the calculations in the spreadsheet will need to be present in your code
- ❑ You are free to use any library to achieve the results, but bear in mind that the calculations needs to happen dynamically every time that the function is called
- ❑ Your Python code **can't** use the spreadsheet as a source, but you can create any other external method as csv, txt, etc.. Since your result is easy reproducible in any other environment
- ❑ You must make available in the **Github** (personal account) and send us the repository link containing the instructions to run your code

NICE TO HAVE

- ❑ Create a docker container to run the code. All the instructions in how to build and run the container needs to be in the README file