

Instructions:

1. Published Date : 15-Aug-2021
2. Submission Date : 23-Aug-2021

DA 201: Introduction to Data Science 3:1

1

Instructions

1. Codes should be original and should not be copied from others, including fellow participants.
2. Plagiarised reports/codes results in zero mark

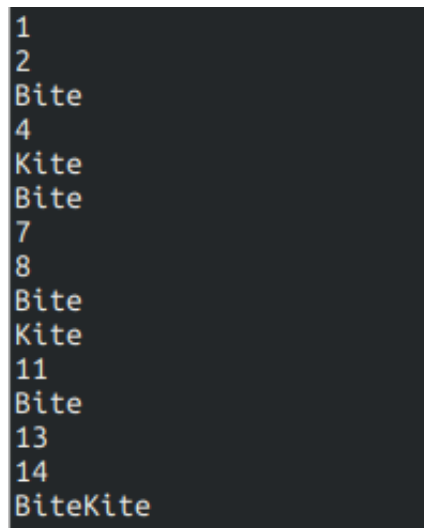
Prob: 1 Bite-Kite

Write a Python program to print numbers from 1 to n . Moreover, an appropriate keyword must be printed instead of the number if the number matches the below criteria. **10**

Points

- if the number is divisible by 3, print "Bite"
- If the number is divisible by 5, print "Kite"
- if the number is divisible by both 3 & 5 (i.e, divisible by 15), print "BiteKite"

For Eg, The output for the problem with $n = 15$ should look like:

A screenshot of a terminal window showing the output of a Python program for n=15. The output is as follows:

```
1
2
Bite
4
Kite
Bite
7
8
Bite
Kite
11
Bite
13
14
BiteKite
```

Figure 1: Sample output for $n=15$

Prob: 2 COVID-19 Analysis

The data file, "KA_COVID.csv " contains the daily COVID-19 dataset for the state of Karnataka. The data contains the number of active cases, number of recovered cases, number of deceased cases recorded in every district of Karnataka from 02-Jul-2020 to 11-Jul-2020.

¹Problem set 1

	Date	Confirmed	Deceased	Recovered
0	2020-06-01	16514	253	8065
1	2020-07-02	1502	19	271
2	2020-07-03	1694	21	471
3	2020-07-04	1839	42	439
4	2020-07-05	1925	37	603

Figure 2: dataframe after being loaded from csv

	Date	Confirmed	Deceased	Recovered	TotalConfirmed	TotalDeceased	TotalRecovered	RatioRecovered
1	2020-07-02	1502	19	271	18016	272	8336	0.180426
2	2020-07-03	1694	21	471	19710	293	8807	0.278040
3	2020-07-04	1839	42	439	21549	335	9246	0.238717
4	2020-07-05	1925	37	603	23474	372	9849	0.313247
5	2020-07-06	1843	30	680	25317	402	10529	0.368964

Figure 3: final dataframe after adding all columns

The first row (01-Jul-20) contains the cumulative total of active, recovered, deceased cases until 01-Jul-20, from the following rows, all the data are the active, confirmed, recovered cases reported as of that day.

Part 2a: Creating Dataframe [20 Points]

1. Read the data from the CSV file into a Pandas dataframe as shown in fig 2
2. Perform a cumulative sum of the Confirmed, Recovered, and Deceased cases and add them as new columns (TotalConfirmed, TotalDeceased, TotalRecovered) to the corresponding dates in the dataframe as shown in fig 3
3. Drop(delete) the first row (01-Jun-2020) from the created dataframe.
4. Compute the ratio of Recovered to Confirmed and add it as a new column with the name "RatioRecovered" to the existing data frame.

Part 2b: obtaining insights from data [15 Points - 5 Points each]

1. Compute the date of the peak for wave-2 (based on daily active cases)
2. Compute the daily active case load (Confirmed) at the date of peak.
3. Compute the mean of the RatioRecovered column in the dataframe

Part 2c: Plotting [15 Points - 5 Points each]

use matplotlib to generate two subplots and plot the following in each of the subplots as line plots.

1. subplot-1: plot a line plot of Confirmed cases, Recovered cases, Deceased cases
2. subplot-2: plot a line plot of TotalConfirmed, TotalRecovered, TotalDeceased cases
3. The plots should have proper labels, legends, and titles. The xticks should be present at first date of every month starting from (Aug-20). The format of the xticklabel should be in the form of Aug'20 as shown in fig 4

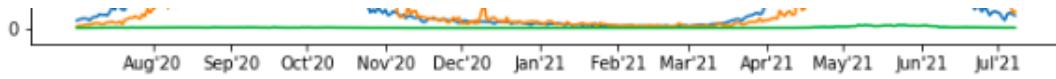


Figure 4: xtick labels for the plot

Submission Instructions

- Problem 1, and Problem 2(a,b) should be written and submitted in the quiz section of moodle page. (Coderunner type quiz)
- Problem 2(c), the generated plot and the code should be submitted as a PDF in the moodle page (Descriptive assignment type)

Common python syntax

- Identify the index at which the maximum value of a column in a dataframe occurs
`df[['Column']].idxmax().item()`
- Identify the maximum value of a column in a dataframe
`df[['Column']].max().item()`
- Compute Cumulative sum of a column
`df["Column"].cumsum()`
- To divide one dataframe column(A) by another column(B)
`df["A"].div(df["B"])`
- To add a column to existing dataframe from a list
`df["NewColumnName"] = newColumnList`
- Drop/Delete n^{th} row from the dataframe
`df = df.drop([n],axis=0)`
- To plot a dataframe with column(X) as x value and column(Y) as y value
`plt.plot(df["X"],df["Y"])`