Graded Questions (Model Selection and Model Evaluation):

1. Model Selection: No Graded Questions
2. Model Evaluation (Practice Questions)

A screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generatedA screenshot of a cell phone

Description automatically generated

1. Coding Graded:

Question 1

**Break the list**

Write a Python program to divide a given list into chunks of size k.  
  
The number of elements in the list need not to be divisible by k.  
For example, if you want to divide the list [1,2,3,4,5,6,7] into chunk size k=4, then the first chunk will be [1,2,3,4] and the second one will have [5,6,7]. i.e. the last chunk need not have k elements.  
The input will have two lines, the first line would have the list and the second line would have the value of k.(the code for taking input has already been written, you should not change that)  
The final output should have the list chunks in different lines.  
  
Sample Input:  
[1,2,3,4,5,6,7,8,9]  
3  
Sample Output:

[1, 2, 3]

[4, 5, 6]

[7, 8, 9]

A similar question was asked in a Data Analyst test.

Solution:

import ast,sys

input\_str = sys.stdin.read()

input\_list = ast.literal\_eval(input\_str)

lis=input\_list[0]

k=input\_list[1]

for item in range(0, len(lis), k):

print(lis[item:item + k])

Question 2

**Scrabble**

Assume that you are trying to complete a crossword puzzle. In a crossword puzzle, some letters are given and you have to figure out which complete word can you make out of it.  
For example, given letters "cwd"in the same order, you can make the word  "crossword" or "crowd" with it. But if "dw" is given in that order, you cannot make "crossword" out of it because d and w are in the opposite order here.  
  
You want to find an efficient method and write a code for this process.   
The code will take two strings as input in two lines. The first string will contain letters on the crossword("cwd"), and the second string will contain a word that you want to check.(i.e. you want to check whether the word can be made out of given letters in the same order).  
The output should say "yes" if the word can be formed, else it should say "no"  
Sample Input 1:  
ccwd  
crossword  
Sample Output1:  
no  
  
Sample Input 2:  
cwd  
crossword  
Sample Output 2:  
yes  
  
**Note:**Using regex is also one of the ways of solving this problem. You can read more about regex(regular expressions) [here.](https://www.w3schools.com/python/python_regex.asp) Although, you can simply use loops and conditionals too to solve the problem.

Solution:

letters=input()#letters already in the crossword

guess=input()#word to check for fit

cnt = 0

for words in guess:

if words == letters[cnt]:

cnt += 1

if cnt >= len(letters): break

if (cnt == len(letters)):

print('yes')

else:

print('no')

Question 3

**Paying the rent**

You are moving to a new city for an internship for a few months and have to rent a house for that purpose.  
You have to pay the full month's house rent even if you have lived for a few days for that month. i.e. if you start on 15th Jan and leave by 15th May, you still have to pay full rent for months of Jan and May too.  
  
Your task is to find the months that you have to pay rent for so that you can write post-dated cheques to your landlord.  
  
You will be given two dates as input and your task is to print all months between the dates including the months the dates are from.  
The input will contain the two dates in a list as follows:  
[2017,1,1,2017,3,4] which corresponds to 1st Jan, 2017 and 4th March, 2017. This date is in the format(yyyy,mm,dd)  
(the code for taking input has already been written for you, please don't modify that)  
The output should contain a list with names of months you have to pay the rent for(the list should be sorted chronologically  based on months, i.e May should come before August). You can assume that there won't be more than 12 months between two dates.  
You'll need to use the datetime module for this problem which can be referred to [here](https://docs.python.org/2/library/datetime.html). **You can choose to use any other method too.**  
  
Sample Input:

[2017,1,1,2017,1,1]

Sample Output:

['January']

This question was asked in a Data Analyst coding test

Solution:

import datetime

import ast,sys

input\_str = sys.stdin.read()

input\_list = ast.literal\_eval(input\_str)

dateStart=datetime.date(input\_list[0],input\_list[1],input\_list[2])

dateEnd=datetime.date(input\_list[3],input\_list[4],input\_list[5])

month\_list = ['January', 'February', 'March', 'April', 'May', 'June', 'July', 'August', 'September', 'October', 'November', 'December']

k = set()

while dateStart <= dateEnd:

k.add(dateStart.month)

dateStart += datetime.timedelta(1)

print([month\_list[x-1] for x in sorted(k)])

Question 4

**Remove empty lists**

Write a Python program to eliminate all empty lists from a given list of lists.  
The input will have a list of lists and the output should have the resulting list after eliminating empty lists from it.  
  
Sample Input:

[[1,2,3],[],[4,5],[],[],[6,7,8]]

Sample Output:

[[1, 2, 3], [4, 5], [6, 7, 8]]

A similar question was asked in a Data Scientist interview coding round.

Solution:

import ast,sys

input\_str = sys.stdin.read()

input\_list = ast.literal\_eval(input\_str)

final\_list = [item for item in input\_list if item != []]

print(final\_list)