Points to keep in mind:

1. **code should work for any general input** `try not to hard code`
2. **Try avoiding use of Numpy and Sklearn**

$$AB = \begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \cdot \begin{bmatrix} 5 & 6 \\ 7 & 8 \end{bmatrix}$$

$$= \begin{bmatrix} 1(5) + 2(7) & 1(6) + 2(8) \\ 3(5) + 4(7) & 3(6) + 4(8) \end{bmatrix}$$

$$= \begin{bmatrix} 19 & 22 \\ 43 & 50 \end{bmatrix}$$

Say dimension of matrix `m_1 is r * c` **and matrix** `m_2 is c * r` **, then dimension of resulting matrix will be** `r × r`

1. Multiply two matrices

```
`Test 1`: m_1   = [[10,88,55]
                   [1,32,56]
                   [53,30,66]]
          m_2   = [[1,0,0]
                   [0,1,0]
                   [0,0,1]]
          m_1 * m_2 = [[10,88,55]
                       [1,32,56]
                       [53,30,66]]

`Test 2`: m_1   = [[11,44]
                   [90,43]]
          m_2   = [[1,2,3,4,5]
                   [66,32,65,34,87]]
          m_1 * m_2 = [[2915, 1430, 2893, 1540, 3883]
                       [2928, 1556, 3065, 1822, 4191]]

`Test 3`: m_1   = [[10 2]
                   [37 44]]
          m_2   = [[1 44]
                   [55 6]
                   [73 85]
                   [91 60]]
          m_1 * m_2 = Not possible
```

- **For two matrices to be multiplied,number of columns in first matrix must be equal to number of rows in second matrix**

You have to write other steps for matric multiplication

In [ ]:

```
# we can also make a function to do this: you have to make a function to do this

m_1 = [[10,88,55],[1,32,56],[53,30,66]]
m_2 = [[1,0,0],[0,1,0],[0,0,1]]


row1 = len(m_1) #row in matric 1
col1 = len(m_1[0])  #columns in matric 1

# calculate number of rows and columns present in second matrix
row2 = len(m_2)
col2 = len(m_2[0])


if(col1 != row2):
    print("Metrices can't be multiplied")
else:
    #array prod will hold result and is initialized with zeroes
    prod = [[0]*row1 for i in range(col2)]

    # performs product of matrices m_1 and m_2
```

```
    # Storing result in matrix prod
    for i in range(0,row1):
        for j in range(0,col2):
            for k in range(0,row2):
                prod[i][j] = prod[i][j] + m_1[i][k] * m_2[k][j]

    print('Product of matrices: ')
    for i in range(0,row1):
        for j in range(0,col2):
            print(prod[i][j])
        print(" ")
```

2. Your result should look same as under given 3 Test sets

```
`Test 1`: m_1   = [[10,88,55]
                   [1,32,56]
                   [53,30,66]]
          m_2  = [[1,0,0]
                  [0,1,0]
                  [0,0,1]]
          m_1 * m_2 = [[10,88,55]
                       [1,32,56]
                       [53,30,66]]


`Test 2`: m_1   = [[11,44]
                   [90,43]]
          m_2   = [[1,2,3,4,5]
                   [66,32,65,34,87]]
          m_1 * m_2 = [[2915, 1430, 2893, 1540, 3883]
                       [2928, 1556, 3065, 1822, 4191]]

`Test 3`: m_1   = [[10 2]
                   [37 44]]
          m_2   = [[1 44]
                   [55 6]
                   [73 85]
                   [91 60]]
          m_1 * m_2 = Not possible
```

As you can see your results are sequential for above code but without these"[]" sign

In [ ]:

3. Data is in string data type if digits found in string replace all with *

consider a string that will have digits in that, we need to remove all not digits and replace digits with *

```
Ex 1: A = 546               Output: ***
Ex 2: A = s5t7r9            Output: ***
Ex 3: A = string            Output:   (empty/nothing)
Ex 5: A = #2a$#b%c%561#     Output: ****
```

In [ ]:

```
import string
str = "#2a$#b%c%561#" # input string
str_new = []
for i in str:
    if i in string.digits:
        str_new.append("*") #add "*" if its a digit else don't do anything

print("".join(str_new))
```

4. Write two valid sentences

Find

a. Number of common words between Sen_1, Sen_2 (and what are they)--Option1a
b. Words which are in Sen_1 but are not in Sen_2
c. Words which are in Sen_2 but are not in Sen_1

In [ ]:

```
def string_diff(Sen_1,Sen_2):
    # without membership operator
    diff1 = list(set(Sen_2.split()) - set(Sen_1.split()))
    diff2 = list(set(Sen_1.split()) - set(Sen_2.split()))

    # with membership operators
    diff1_1 = [each_word for each_word in Sen_1.split() if each_word not in Sen_2.split()]
    diff2_1 = [each_word for each_word in Sen_2.split() if each_word not in Sen_1.split()]
    common_words = [each_word for each_word in Sen_2.split() if each_word in Sen_1.split()]
    common_words_count = len(common_words)

    print(f'Words in Sen_1 but not in Sen_2 are : {diff1}')
    print(f'Words in Sen_1 but not in Sen_2 are : {diff2}','\n','***'*10)
    print(f'Words in Sen_1 but not in Sen_2 are : {diff1_1}')
    print(f'Words in Sen_2 but not in Sen_1 are : {diff2_1}')
    print(f'Common Words in Sen_1 and Sen_2 are : {common_words}')
    print(f'Count of Common Words in Sen_1 and Sen_2 is : {common_words_count}')

Sen_1 = 'You are the real magic Xgboost and NN are algos'
Sen_2 = 'You are the real magic GBboost and RNN are algos'
string_diff(Sen_1,Sen_2)
```

5. Copy a list using function

Copying list means both variable must be unique objects

- In general you can take 3 approaches to do this
  - Approach used in Python 2.x => write 2 here ==> .....(slicing,list() func)
  - Approach used in Python 3.x => write 1 here ==> .....(copy())
- After making copy of list technically prove that it's a copy

```
list_1 = [1,2,3,4,5,6]
list_2 = copy_list(list_1)
```

```
Original List: [1, 2, 3, 4, 5, 6]
Copy List: [1, 2, 3, 4, 5, 6]

Prove-1: as both variables have different id's : 139957125261264---139957125582432

Prove-2: as now original list_2 is: [10, 2, 3, 4, 5, 6] and change in list_2--does not effect list_1: [1, 2, 3, 4, 5, 6]
```

In [ ]:

```
def copy_list(list_1):
    list_copy = list_1[:] #use of slicing
    return list_copy

list_1 = [1,2,3,4,5,6]
list_2 = copy_list(list_1)

print(f'Original List: {list_1}')
print(f'Copy List: {list_2}','\n')

print(f"Prove-1: as both variables have different id's : {id(list_1)}---{id(list_2)}",'\n')
# changing list 2 value at index 0
list_2[0] = 10
print(f"Prove-2: as now original list_2 is: {list_2} and change in list_2--does not effect list_1: {list_
```

Now after solving this again go to concept of Deep and Shallow copy

6. Adding nested lists(metrices) Perpendicularly

With Constraint

- You can only use the variable named as `given_list`

Output:

```
Given list : [['Welcome', 'You'], [' to', ' can'], [' cloudyML.', ' do this']]
List after column Concatenation :['Welcome to cloudyML.', 'You can do this']
```

In [ ]:

```
given_list = [['Welcome','You'], [' to',' can'], [' cloudyML.',' do this']] # shape is (3,2)
print('Given list : '+str(given_list))


res = []
N = 0
while N != len(given_list):
    temp = ''
    for idx in given_list:
        try: temp = temp + idx[N]
        except IndexError: pass
    res.append(temp)
    N = N + 1

res = [ele for ele in res if ele]

print('List after column Concatenation :' + str(res))
```

```
Given list : [['Welcome', 'You'], [' to', ' can'], [' cloudyML.', ' do this']]
List after column Concatenation :['Welcome to cloudyML.', 'You can do this']
```

7. Remove empty List from List of List's

Write a function to remove Empty List from a given nested list:

Output:

```
given_list = [90,634,[],3343,[],['Mukesh Manral'],[],956343,['CloudyML']]
empty_list_remover(given_list)
```

```
[90, 634, 3343, ['Mukesh Manral'], 956343, ['CloudyML']]
```

In [ ]:

```
def empty_list_remover(given_list):
    given_list_modefyied = [element for element in given_list if element != []]
    return given_list_modefyied

given_list = [90,634,[],3343,[],['Mukesh Manral'],[],956343,['CloudyML']]
empty_list_remover(given_list)
```

Out[ ]:

```
[90, 634, 3343, ['Mukesh Manral'], 956343, ['CloudyML']]
```

8. Write your explaination on diff. of for and while loop for a 10 year old kid

Comment here.....

In reallife interviews you will be given code and you will be asked to tell or select the output of the code
From here onwards I am trying to simulate reallife Python interview Questions

Enjoy... You can do this

What you have to do:

1. Read code
2. Map it into your brain
   - If you can't understand flow of the code go to Python video, I have mentioned website to see flow of code
3. Understand the output, recomplile it into your brain
4. Then see if there are any kind of Error in the code
   - Say `NameError` , `KeyError` and etc
5. Resolve Error if found
6. Run code
7. Explain flow of the code
8. Explain why the output

9. Resolve Error and find out What will be output of this code

- Understand problem
- Map it
- Come with a logical reasoning
- Validate your answer by running this code

1. Resolve the Error bellow

2. Tell your answer for value's of number as:

- 12 ---> your answer first
- 23 ---> your answer first
- 2 ---> your answer first
- 6 ---> your answer first

Output:
For 12 --> 14 ,23 --> 25, 2 --> 0, 6 --> 8

In [ ]:

```
def process_1(number):
    if number < 6:
        return number - 1
    else:
        return number + 1
def process_2(number):
    if number > 6:
        return number + 1
    else:
        return process_1(number)

process_2(process_1(number))
```

10. Predict Output

What will be output if we give key as Bharat to given dict?

Output:
1

In [ ]:

```
global_ranking = {
    'Bharat': 50,
    'China' : 10,
    'Pakistan': 1000,
    'USA':1,
    'Bharat':1
}
```

11. Predict Output

Remove that error line

- Write your answer!! .....
- Explain What is going on inside this code?? .....
- When if elif and else condition will be True one after other Explain?? .....

Crude Output:

```
Alas I dont know Bhagavad Gita
```

In [ ]:

```
if len({1,True}) >= 4:
    print('I love reading Bhagavad Gita')
elif len({1,False,0,True}) > 2:
    print('Gem of all')
else:
    print('Alas I dont know Bhagavad Gita')

gibrishhhh
```

12. Predict Output

Remove Error line before that imagin Answer

- Explain why is the output the way it is???

Crude Output:

```
'Chanting of mantras'
```

In [ ]:

```
chek_dict_knowledge = {
    101:'Mantras',
    108:'Chanting',
    '101':'Chanting of mantras'
}
chek_dict_knowledge['101']

print Error
```

13. Predict Output

- What will be the output ??
- Write your explanation ??

Output:

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_33/4096844962.py in <module>
      1 set_knowledge_check = {11,21,51,101,108,111}
----> 2 set_knowledge_check.add(['chant','mantras'])

TypeError: unhashable type: 'list'
```

In [ ]:

```
set_knowledge_check = {11,21,51,101,108,111}
set_knowledge_check.add(['chant','mantras'])
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
<ipython-input-5-76ba8a01f4c9> in <module>()
      1 set_knowledge_check = {11,21,51,101,108,111}
----> 2 set_knowledge_check.add(['chant','mantras'])

TypeError: unhashable type: 'list'
```

14. Predict Output

Remove Error causing line

- What will be the output ??
- Write your explanation!!

Output:

```
{11, 21, 51, 101, 108, 111}
```

In [ ]:

```
set_knowledge_check_1 = {11,21,51,101,108,111}
set_knowledge_check_2 = {111,11,21,51,101,108,1008}

set_knowledge_check_2 = set_knowledge_check_2.intersection(set_knowledge_check_1)
peinr(set_knowledge_check_2)
```

15. Predict Output

- What will be the output
- Write your explanation

Output:

```
(11, 21, 51, 'chant', 'mantras')
```

In [ ]:

```
tuple_knowledge_check = (11,21,51,'chant','mantras',101,108,111)
type(tuple_knowledge_check[0:5])
```

16. Predict Output

- Write different ways to get 'chant' as an answer

Output:

These all code lines will give chant as output ther might be other pnc

python_datatype_knowledge_check[0][1][1]

python_datatype_knowledge_check[1][1][2]

python_datatype_knowledge_check[1][2][0] # and many more maybe

```
'chant'
```


In [ ]:

```
python_datatype_knowledge_check = [
    [{11,21,51},('chant','chant','mantras'),[21,[11,21]]],
    [{101,108,111},('i','we','chant'),['chant','mantras',[21]]],
    [{51,'chant','chant'},('they','there','them'),['21','11','chant']]
]
```

17. Predict Output

- Explain the error and resolve it
- Share your thoughts on Error

Output:

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
/tmp/ipykernel_33/3991384593.py in <module>
      1 easy_one = [11,21,51]
----> 2 print(''.join(easy_one))

TypeError: sequence item 0: expected str instance, int found
```

In [ ]:

```
easy_one = [11,21,51]
print(''.join(easy_one))
```

18. Predict Output

Remove error line

- Imagin answer
- Explain program flow

Output:

```
0
4
2
6
```

In [ ]:

```
initial = 0
while initial < 3:
    print(initial)
    initial += 2
    print(initial + 2)
Error
```

```
0
4
2
6
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-ad8a103c77a9> in <module>()
      4     initial += 2
      5     print(initial + 2)
----> 6 Error

NameError: name 'Error' is not defined
```

By now you must understand flow of program and there output. Lets do some more coding now

19. I want Output

Take First Name and Last Name of a user and print welcome message as : Welcome in ClousyML First name Second name

In [ ]:

```
first_name = input('Please type your First Name here:')
last_name = input('Please type your Last Name here:')
full_name = first_name + last_name

print(f'Welcome in CloudyML {full_name}')
```

```
Please type your First Name here:Mukesh
Please type your Last Name here:Manral
Welcome in CloudyML Mukesh Manral
```

20. I want Output

Take a valid input which must be comma seperated(atlest 4 comma) and change input into Tuple and List

- Print tuple
- print list
- print input original

In [ ]:

```
taken_input = input('Please give comma seperated input')

taken_input_list = taken_input.split(',')
taken_input_tuple = tuple(taken_input_list)

print('**'*10)
print(f'Input changed into tuple: {taken_input_tuple}')
print(f'Input changed into list: {taken_input_list}')
```

```
print(f'Original Input: {taken_input}')

Please give comma seperated input"Mukesh","Manral",1,2,{3,4},[5,6,7]
********************
Input changed into tuple: ('"Mukesh"', '"Manral"', '1', '2', '{3', '4}', '[5', '6', '7]')
Input changed into list: ['"Mukesh"', '"Manral"', '1', '2', '{3', '4}', '[5', '6', '7]']
Original Input: "Mukesh","Manral",1,2,{3,4},[5,6,7]
```

21. I want Output

Point to keep in brain:

Before coding any pattern problem

1. **Outer loop decides Rows number**
2. **Inner loop decides Column number**
3. **For every Inner loop, Outer loop executes completelly**

In [ ]:

```
rows = 11
initial = 1

while initial <= rows:
    j = rows
    while j > initial:
        print(' ', end=' ')
        j -= 1
    print('?', end=' ')

    tough_hai_na_ye = 1
    while tough_hai_na_ye < 2 * (initial-1):
        print(' ', end=' ')
        tough_hai_na_ye += 1
    if initial == 1:
        print()
    else:
        print('?')
    initial += 1

initial = rows - 1
while initial >= 1:
    j = rows
    while j > i:
        print(' ', end=' ')
        j -= 1
    print('?', end=' ')

    learn_by_doing = 1
    while learn_by_doing <= 2 * (initial-1):
        print(' ', end=' ')
        learn_by_doing += 1
    if initial == 1:
        print()
```

```
    else:
        print('?')
    initial -= 1
```

```
                    ?
                  ?   ?
                ?       ?
              ?           ?
            ?               ?
          ?                   ?
        ?                       ?
      ?                           ?
    ?                               ?
  ?                                   ?
?                                       ?
          ?                                       ?
          ?                                   ?
          ?                                 ?
          ?                               ?
          ?                           ?
          ?                       ?
          ?                   ?
          ?               ?
          ?           ?
          ?       ?
          ?
```

22. I want Output

```
                    .
                   ..
                  ...
                 ....
                .....
               ......
              .......
             ........
            .........
           ..........
      ............
      ............
      ............
      ............
      ............
      ............
      ............
      ............
      ............
      ............
      ............
```

In [ ]:

```
n = 11

for i in range(n):
    for j in range(n - i - 1):
        print(' ', end=' ')
    for k in range(i + 1):
        print('.', end='')
    print()

for i in range(n - 1):
    for j in range(i + 1):
        print('.', end='')
    for k in range(n - i - 1):
        print('.', end='')
    print()
```

In [ ]: