



Take-Home (Day 3)

Let's begin with some hands-on practice exercises



1. Write a program to print the following pattern (use for loop)

```
A
B C
D E F
G H I J
K L M N O
P Q R S T U
```

```
In [19]: # type your code here
def contalpha(n):
    #Initiating the ascii value of A
    num = 65
    for i in range(0, n):
        for j in range(0, i+1):
            ch = chr(num)
            print(ch, end=" ")
            num = num +1
        print("\r")
    n = int(input("Enter the number of rows: "))
    contalpha(n)
```

Enter the number of rows: 6

```
A
B C
D E F
G H I J
K L M N O
P Q R S T U
```



2. Write a program to print tables from 2 to a user provided range (use for loop)

```
In [12]: # type your code here
n = int(input("Enter the range for the 2's tabel: "))
for i in range(1,n+1):
    print("2*{}={}".format(i,2*i))
```

Enter the range for the 2's tabel: 13

2*1=2

2*2=4

2*3=6

2*4=8

2*5=10

2*6=12

2*7=14

2*8=16

2*9=18

2*10=20

2*11=22

2*12=24

2*13=26



3. Consider the given two tuples. Divide corresponding elemets of two tuples and print the result as tuple (use zip())

The original tuple 1 : (10, 4, 6, 9)

The original tuple 2 : (5, 2, 3, 3)

Output: (2, 2, 2, 3)

```
In [13]: # type your code here
tup1 = (10, 4, 6, 9)
tup2 = (5, 2, 3, 3)
res_tup = tuple(i1//i2 for i1,i2 in zip(tup1,tup2))
print(res_tup)
```

(2, 2, 2, 3)



4. Use list comprehension to find words whose length is less than 5 from the given multi-dimensional list

astro = [['sky', 'sun', 'moon'], ['star', 'Jupiter', 'mars'], ['Jupiter', 'Neptune', 'Pluto']]

```
In [20]: # type your code here
astro = [['sky', 'sun', 'moon'], ['star', 'Jupiter', 'mars'], ['Jupiter', 'Nep
tune', 'Pluto']]
res_lst = [astro[i][j] for i in range(len(astro)) for j in range(len(astro[i
])) if int(len(astro[i][j]))<5]
print(res_lst)

['sky', 'sun', 'moon', 'star', 'mars']
```

5. Write a program to append a dictionary to a tuple

The original tuple : (4, 5, 6)

Tuple after addition of dictionary : (4, 5, 6, {'gfg': 1, 'is': 2, 'best': 3})

```
In [14]: # type your code here
tup = (4, 5, 6)
dit = (4, 5, 6, {'gfg': 1, 'is': 2, 'best': 3})
print("The original tuple is : ",tup)
print("The original dictionary is : ",dit)
t_tup = list(tup)
t_tup.append(dit)
tup = tuple(t_tup)
print("The tuple after appending the dictioanry is : ",tup)
```

```
The original tuple is : (4, 5, 6)
The original dictionary is : (4, 5, 6, {'gfg': 1, 'is': 2, 'best': 3})
The tuple after appending the dictioanry is : (4, 5, 6, (4, 5, 6, {'gfg': 1,
'is': 2, 'best': 3}))
```

6. Write a program to check whether two strings are anagram or not (take input from user)

Hint: An anagram of a string is another string that contains same characters, only the order of characters can be different.

```
In [15]: # type your code here
#Anagram words listen = silent, triangle = integral
def areAnagram(str1, str2):
    n1 = len(str1)
    n2 = len(str2)

    if n1 != n2:
        return 0

    str1 = sorted(str1)
    str2 = sorted(str2)

    for i in range(0, n1):
        if str1[i] != str2[i]:
            return 0

    return 1

str1 = str(input("Enter string 1 : "))
str2 = str(input("Enter string 2 : "))

if areAnagram(str1, str2):
    print("The two strings are anagram of each other")
else:
    print("The two strings are not anagram of each other")
```

```
Enter string 1 : silent
Enter string 2 : listen
The two strings are anagram of each other
```



7. Write a program to remove space from keys of the given dictionary (use for loop)

```
fruity = {'Fruits 01' : 'Apple', 'Fruits 02' : 'Orange', 'Fruits 0 3 ' : 'Grapes'}
```

(Output) New dictionary : {'Fruits01': 'Apple', 'Fruits02': 'Orange', 'Fruits03': 'Grapes'}

```
In [24]: # type your code here
fruity = {'Fruits 01' : 'Apple', 'Fruits 02' : 'Orange', 'Fruits 0 3 ' : 'Grapes'}
fruity = {x.replace(' ', ''): v
          for x, v in fruity.items()}
print (" New dictionary : ", fruity)

New dictionary :  {'Fruits01': 'Apple', 'Fruits02': 'Orange', 'Fruits03': 'Grapes'}
```



8. Write a program to transpose a matrix using list comprehension

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

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

```
In [23]: # type your code here
matrix = [[1,2],[3,4],[5,6],[7,8]]
print("The origianl matrix is : ")
for row in matrix :
    print(row)
rez = [[matrix[j][i] for j in range(len(matrix))] for i in range(len(matrix[0]))]
print("The transpose matrix is :")
for row in rez:
    print(row)
```

The origianl matrix is :

[1, 2]

[3, 4]

[5, 6]

[7, 8]

The transpose matrix is :

[1, 3, 5, 7]

[2, 4, 6, 8]



9. Write a program to add the cubes of first 5 positive integers (use while loop)

```
In [16]: # type your code here
sum = 0
i = 1
while(i<=5):
    sum+=i**3
    i+=1
print("The sum of the 1st 5 positive integers is : ",sum)
```

The sum of the 1st 5 positive integers is : 225



10. Write a program to create a 5x3 matrix with values in each row as 1,2, and 3 (use list comprehension)

```
In [25]: # type your code here
mat1 = [[1,2,3] for i in range(5)]
print(mat1)
```

[[1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3], [1, 2, 3]]