

DAY 3 ASSIGNMENT:

infyTQ | Viewer

infytq.infosys.com/viewer/lex_auth_0125409616243425281061?filterCategory=Assess

LEARNING JOURNEY CERTIFICATION ALUMNI STORIES CONTEST ARENA GALLERY GET TO KNOW US

PROGRAMMING FUNDAMENTALS USING PYTHON

Assignment 22: Collaborative Assignment - Level 2

Problem Statement Starter Testcase Submission

Code pane:

```
1 #PF-Assgn-22-
2 def find_leap_years(given_year):~
3 ~
4 ~# Write your logic here~
5 ~c=15
6 ~l=[]
7 ~while c!=0:~
8 ~~~~~if given_year%4==0 or given_year%400==0 and given_year%100!=0 :~
9 ~~~~~~l.append(given_year)~
10 ~~~~~~given_year+=1~
11 ~~~~~~c-=1~
12 ~~~~~~return l~
13 ~~~~~~
14 ~~~~~~
15 ~~~~~~return list_of_leap_years~
16 ~~~~~~
```

Verify Submit Python Visualize Execute

Result Pane:

[2000, 2004, 2008, 2012, 2016, 2020, 2024, 2028, 2032, 2036, 2040, 2044, 2048, 2052]

Your code has been submitted.

infyTQ | Viewer

infytq.infosys.com/viewer/lex_auth_0125409616243425281061?filterCategory=Assess

LEARNING JOURNEY CERTIFICATION ALUMNI STORIES CONTEST ARENA GALLERY GET TO KNOW US

PROGRAMMING FUNDAMENTALS USING PYTHON

Assignment 23: Collaborative Assignment - Level 2

Code pane:

```
3 ~~~~~bill_amount=0~
4 ~~~~~#Write your logic here~
5 ~~~~~for i in reqd_gems:~
6 ~~~~~~if i not in gems_list:~
7 ~~~~~~bill_amount=-1~
8 ~~~~~~else:~
9 ~~~~~~for i in reqd_gems:~
10 ~~~~~~ind=gems_list.index(i)~
11 ~~~~~~ind2=reqd_gems.index(i)~
12 ~~~~~~bill_amount=bill_amount+int(price_list[ind])*int~
13 ~~~~~~(reqd_quantity[ind2])~
14 ~~~~~~return bill_amount~
15 ~~~~~~
16 ~~~~~~#List of gems available in the store~
17 ~~~~~~gems_list=["Emerald","Ivory","Jasper","Ruby","Garnet"]~
18 ~~~~~~
```

Verify Submit Python Visualize Execute

Result Pane:

71945

Your code has been submitted.

DAY 3 ASSIGNMENT:

The screenshot shows the InfyTQ Viewer interface. The top navigation bar includes links for LEARNING JOURNEY, CERTIFICATION, ALUMNI STORIES, CONTEST ARENA, GALLERY, and GET TO KNOW US. The main content area is divided into three tabs: Problem Statement, Starter Testcase, and Submission. The Submission tab is active, displaying a code editor with the following Python code:

```
1 #Do not change the messages provided below-
2 success="Triangle can be formed"
3 failure="Triangle can't be formed"
4
5
6
7 #Write your logic here-
8 if num1>num2+num3 or num2>num1+num3 or num3>num2+num1:-
9     print(failure)
10 else:-
11     print(success)
12 #Use the following messages to return the result wherever necessary-
13
14
15
16 #Provide different values for the variables, num1, num2, num3 and test
   your program-
17 num1=3
```

Below the code editor are buttons for Verify, Submit, Visualize, and Execute. The Result Pane shows the output: "Triangle can be formed". A purple banner at the bottom states "Your code has been submitted." The footer includes links for Contact Us, FAQs, Feedback, and Terms and Conditions, along with the copyright notice "© 2020, Infosys Ltd." and the system clock showing 21:31 on 23-06-2020.

The screenshot shows the InfyTQ Viewer interface with the same Python code as the previous screenshot. The Result Pane displays the following test results:

9 out of 9 test cases passed.
1 out of 1 structural test cases passed.
8 out of 8 logical test cases passed.

S/A	Type	Sl.no	Test Target	Input	Expected	Actual	Result
N/A	Structural	1	solve(heads, legs)	N/A	N/A	N/A	✓

The footer includes links for Contact Us, FAQs, Feedback, and Terms and Conditions, along with the copyright notice "© 2020, Infosys Ltd." and the system clock showing 21:56 on 23-06-2020.

DAY 3 ASSIGNMENT:

The image displays two screenshots of the InfyTQ Learning Journey interface, showing programming assignments and their execution results.

Top Screenshot: Assignment 27 - Level 2

The assignment is titled "Assignment 27 - Level 2" with an estimated time of 15 minutes. It includes a "Run Code" button. The code provided is as follows:

```
8
9 alex.color("green") # alex has a color
10 alex.right(60)
11 #draws circles
12 for counter in range(1,5):
13     alex.circle(20*counter)
14
15 alex.color("red") # alex has a color
16 alex.right(120)
17 #draws circles
18 for counter in range(1,5):
19     alex.circle(20*counter)
20
21 alex.color("blue") # alex has a color
22 alex.right(120)
23 #draws circles
24 for counter in range(1,5):
25     alex.circle(20*counter)
26
27
28 #Write the logic to create the given pattern
29 #Refer the statements given above to draw the patt
30
```

The visual output shows a pattern of overlapping circles in green, red, and blue, arranged in a symmetrical, flower-like shape.

Bottom Screenshot: Exercise 28 - Level 2 python prog

The assignment is titled "Exercise 28 - Level 2 python prog". It includes a "Submit" button. The code provided is as follows:

```
Code pane:
10 ..... for j in num:-
11 .....     s+=int(j)-
12 .....     if s%3==0 and i%5==0 and len(num)==2:-
13 .....         l.append(i)
14 .....     if len(l)>1:-
15 .....         return max(l)-
16 .....     elif len(l)==1:-
17 .....         return l[0]-
18 .....     elif len(l)==0:-
19 .....         return max_num-
20 .....
21 ~
22 #Provide different values for num1 and num2 and test your program.-
23 max_num=find_max(10,15)-
24 print(max_num)
```

The result pane shows the following output:

```
Result Pane:
7 out of 8 test cases passed.
1 out of 1 structural test cases passed.
```

The interface also includes a sidebar with a "Table of Contents" and a "View Course ToC" button. The bottom of the interface shows a footer with "© 2020, Infosys Ltd." and a "Feedback" button.