

ASSIGNMENT -1

PYTHON PROGRAMMING

TEAM ID	PNT2022TMID44357
STUDENT NAME	SWASTHIKA.M.S
STUDENT ROLL NO	731119106035
PROJECT DOMAIN	INTERNET OF THINGS
PROJECT TITLE	IoT BASED SMART CROP PROTECTION SYSTEM FOR AGRICULTURE
DATE	19 SEPTEMBER 2022
MAXIMUM MARKS	2 MARKS

QUESTION-1:

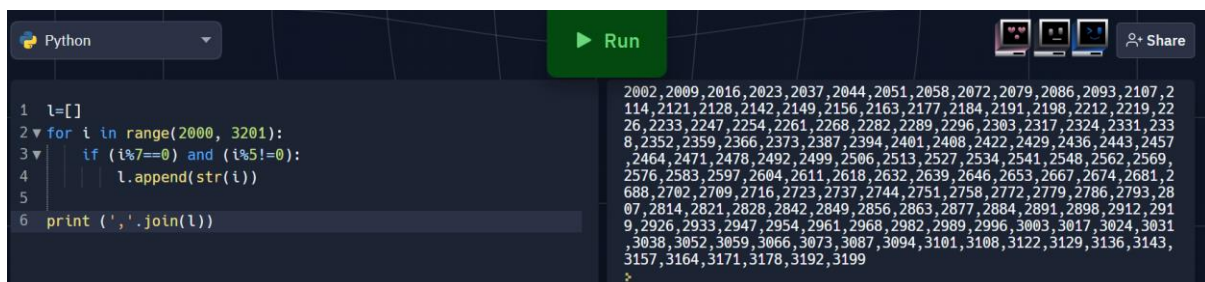
Write a program which will find all such numbers which are divisible by 7 but are not a multiple of 5, between 2000 and 3200 (both included). The numbers obtained should be printed in a comma-separated sequence on a single line.

SOLUTION:

```
l=[]
for i in range(2000, 3201):
    if (i%7==0) and (i%5!=0):
        l.append(str(i))

print(','.join(l))
#-----#
#-----#
```

OUTPUT:



The screenshot shows a Python IDE with a dark theme. On the left, the code is written in a syntax-highlighted editor. On the right, the output of the program is displayed, showing a long list of numbers separated by commas. The code is as follows:

```
1 l=[]
2 for i in range(2000, 3201):
3     if (i%7==0) and (i%5!=0):
4         l.append(str(i))
5
6 print(','.join(l))
```

The output is a single line of text containing all the numbers found by the program, separated by commas. The numbers start at 2002 and end at 3199, with some numbers missing (e.g., 2009, 2016, 2023, etc.).

QUESTION-2:

With a given integral number n, write a program to generate a dictionary that contains (i, i*i) such that i is an integral number between 1 and n (both included). and then the program should print the dictionary.

Suppose the following input is supplied to the program:
8

Then, the output should be:

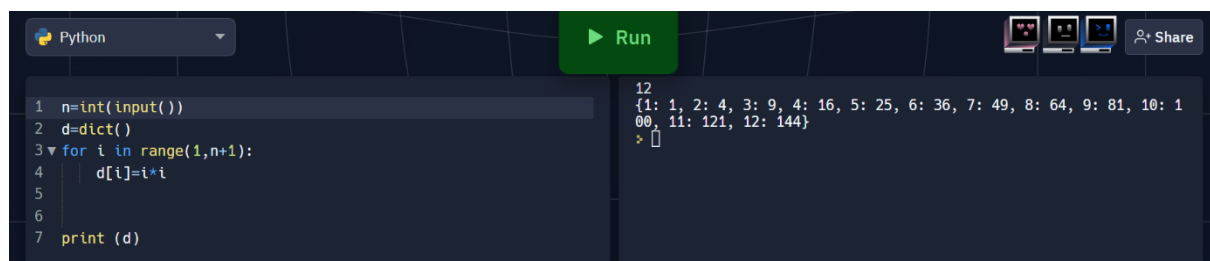
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64}

SOLUTION:

```
n=int(input())
d=dict()
for i in range(1,n+1):
    d[i]=i*i

print d
#-----#
#-----#
```

OUTPUT:



The screenshot shows a Python IDE with a dark theme. The code editor on the left contains the following code:

```
1 n=int(input())
2 d=dict()
3 for i in range(1,n+1):
4     d[i]=i*i
5
6
7 print (d)
```

A green 'Run' button is visible above the code editor. The output console on the right shows the result of running the code:

```
12
{1: 1, 2: 4, 3: 9, 4: 16, 5: 25, 6: 36, 7: 49, 8: 64, 9: 81, 10: 100, 11: 121, 12: 144}
>
```