

## Assignment -1

### Python Programming

Assignment Date	19 September 2022
Team id	PNT2022TMID44153
Student Name	MUHAMMED SANHAR KT
Student Roll Number	704019104015
Maximum Marks	2 Marks

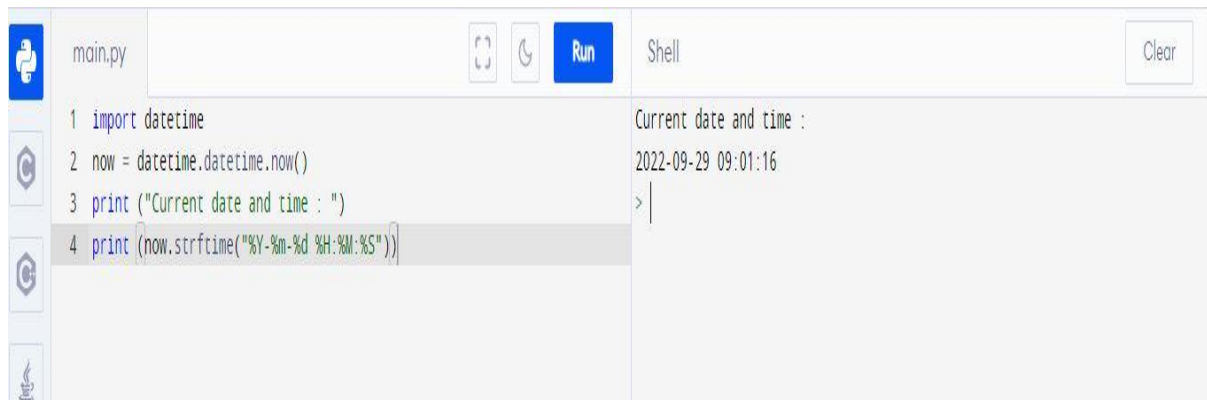
#### Question 1:

Write a Python program to display the current date and time

#### Solution:

```
import datetime
now = datetime.datetime.now()
print ("Current date and time : ")
print (now.strftime("%Y-%m-%d %H:%M:%S"))
```

#### Output:

A screenshot of a Python IDE interface. On the left, a file named 'main.py' is open, containing four lines of Python code: 1. 'import datetime', 2. 'now = datetime.datetime.now()', 3. 'print ("Current date and time : ")', and 4. 'print (now.strftime("%Y-%m-%d %H:%M:%S"))'. The fourth line is highlighted. On the right, the 'Shell' window shows the output of the program: 'Current date and time :', '2022-09-29 09:01:16', and a prompt '> |'. There are icons for running and clearing the shell.

#### Question 2:

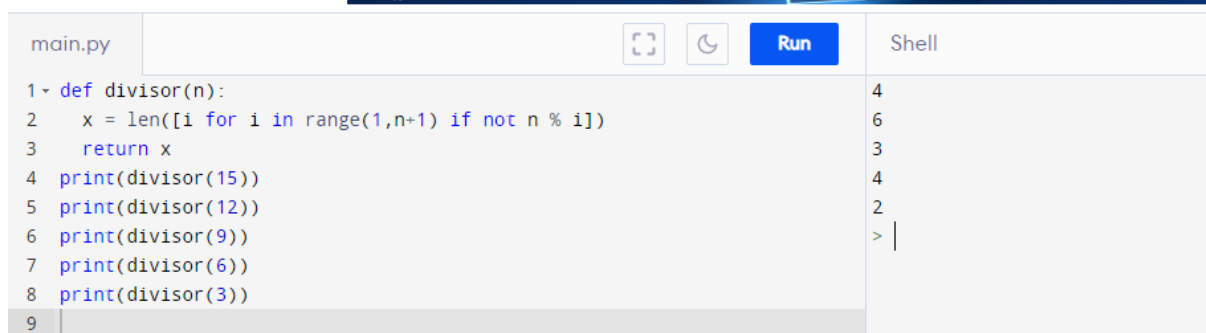
Write a Python program to find the number of divisors of a given integer is even or odd.

#### Solution:

```
def divisor(n):
    x = len([i for i in range(1,n+1) if not n % i])
    return x
print(divisor(15))
```

```
print(divisor(12))  
print(divisor(9))  
print(divisor(6))  
print(divisor(3))
```

### Output:



The screenshot shows a Python IDE with a file named `main.py`. The code in the editor is as follows:

```
1 def divisor(n):  
2     x = len([i for i in range(1,n+1) if not n % i])  
3     return x  
4 print(divisor(15))  
5 print(divisor(12))  
6 print(divisor(9))  
7 print(divisor(6))  
8 print(divisor(3))  
9
```

The output in the shell window is:

```
4  
6  
3  
4  
2  
> |
```

### Question 3:

Write a Python program to compute the amount of the debt in  $n$  months. The borrowing amount is \$100,000 and the loan adds 5% interest of the debt and rounds it to the nearest 1,000 above month by month.

### Input:

An integer  $n$  ( $0 \leq n \leq 100$ )

Input number of months: 7

Amount of debt: \$144000

### Solution:

```
def round_n(n):  
  
    if n%1000:  
  
        return (1+n//1000)*1000  
  
    else:  
  
        return n
```

```
def compute_debt(n):

    if n==0: return 100000

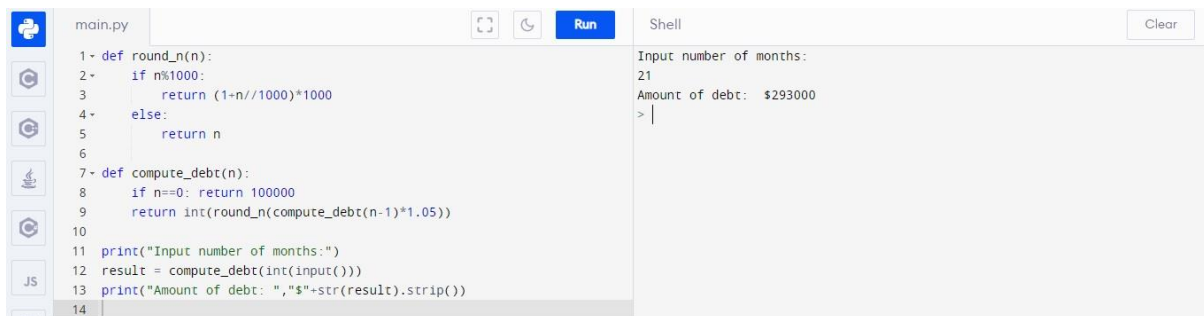
    return int(round_n(compute_debt(n-1)*1.05))

print("Input number of months:")

result = compute_debt(int(input()))

print("Amount of debt: ", "$"+str(result).strip())
```

### Output:



The screenshot shows a Python IDE with a file named 'main.py'. The code in the editor is as follows:

```
1- def round_n(n):
2-     if n%1000:
3-         return (1+n//1000)*1000
4-     else:
5-         return n
6-
7- def compute_debt(n):
8-     if n==0: return 100000
9-     return int(round_n(compute_debt(n-1)*1.05))
10-
11- print("Input number of months:")
12- result = compute_debt(int(input()))
13- print("Amount of debt: ", "$"+str(result).strip())
14-
```

The output in the Shell window is:

```
Input number of months:
21
Amount of debt:  $293000
> |
```




### Question 4:

Write a Python program to check whether a given integer is a palindrome or not.

#### Solution:

```
def is_Palindrome(n):
    return str(n) == str(n)[::-1]
print(is_Palindrome(100))
print(is_Palindrome(252))
print(is_Palindrome(-838))
```

### Output:

main.py	  	Shell
<pre>1 def is_Palindrome(n): 2     return str(n) == str(n)[::-1] 3 print(is_Palindrome(100)) 4 print(is_Palindrome(252)) 5 print(is_Palindrome(-838)) 6</pre>		<pre>False True False &gt;  </pre>

### Question 5:

Write a Python program to find the starting and ending position of a given value in a given array of integers, sorted in ascending order.

### Solution:

```
def search_Range(array_nums, target_val):
    result_arra = []
    start_pos = 0
    end_pos = 0
    for i in range(len(array_nums)):
        if target_val == array_nums[i] and start_pos == -1:
            start_pos = i
            end_pos = i
        elif target_val == array_nums[i] and start_pos != -1:
            end_pos = i
    result_arra.append(start_pos)
    result_arra.append(end_pos)
    return result_arra

print(search_Range([5, 7, 7, 8, 8, 8], 8))
print(search_Range([1, 3, 6, 9, 13, 14], 4))
print(search_Range([5, 7, 7, 8, 10], 8))
```

### Output:

main.py		Shell
1	def search_Range(array_nums, target_val):	[0, 5]
2	result_arra = []	[0, 0]
3	start_pos = 0	[0, 3]
4	end_pos = 0	>
5	for i in range(len(array_nums)):	
6	if target_val == array_nums[i] and start_pos == -1:	
7	start_pos = i	
8	end_pos = i	
9	elif target_val == array_nums[i] and start_pos != -1:	
10	end_pos = i	
11	result_arra.append(start_pos)	
12	result_arra.append(end_pos)	
13	return result_arra	
14	print(search_Range([5, 7, 7, 8, 8, 8], 8))	
15	print(search_Range([1, 3, 6, 9, 13, 14], 4))	
16	print(search_Range([5, 7, 7, 8, 10], 8))	