

## Assignment-1

### Python Programming

Assignment Date	03.10.2022
Student Name	JUWAIIRIYA.M.F
Team id	PNT2022TMID44156
Student Roll Number	724019104007
Maximum Marks	2 MARKS

#### Question-1:

Write a python program to convert decimal to hexadecimal.

Sample decimal number: 30, 4

Expected output: 1e, 04

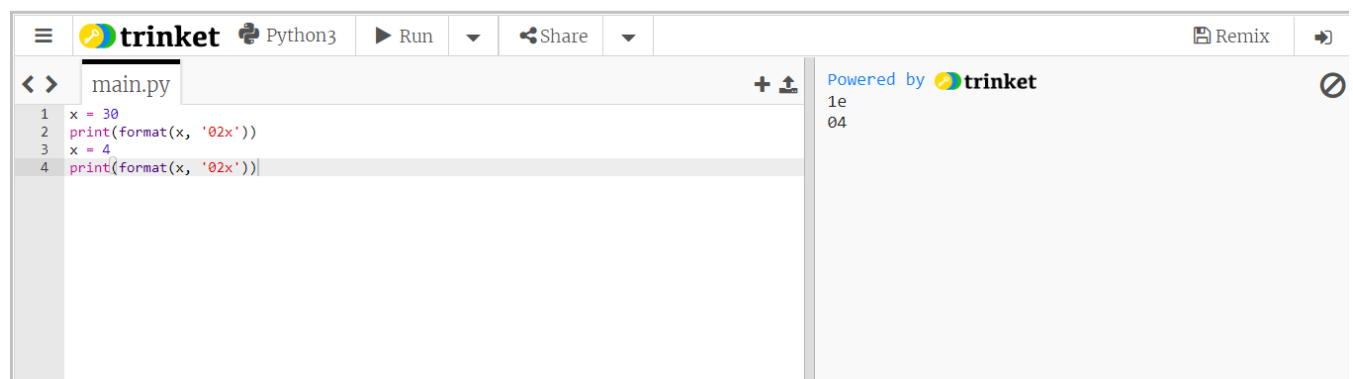
#### Solution:

```
x = 30
```

```
print(format(x, '02x'))
```

```
x = 4 print(format(x, '02x'))
```

#### Output:



The screenshot shows a Trinket Python3 IDE interface. The code editor on the left contains the following code in a file named 'main.py':

```
1 x = 30
2 print(format(x, '02x'))
3 x = 4
4 print(format(x, '02x'))
```

The output panel on the right shows the results of the code execution:

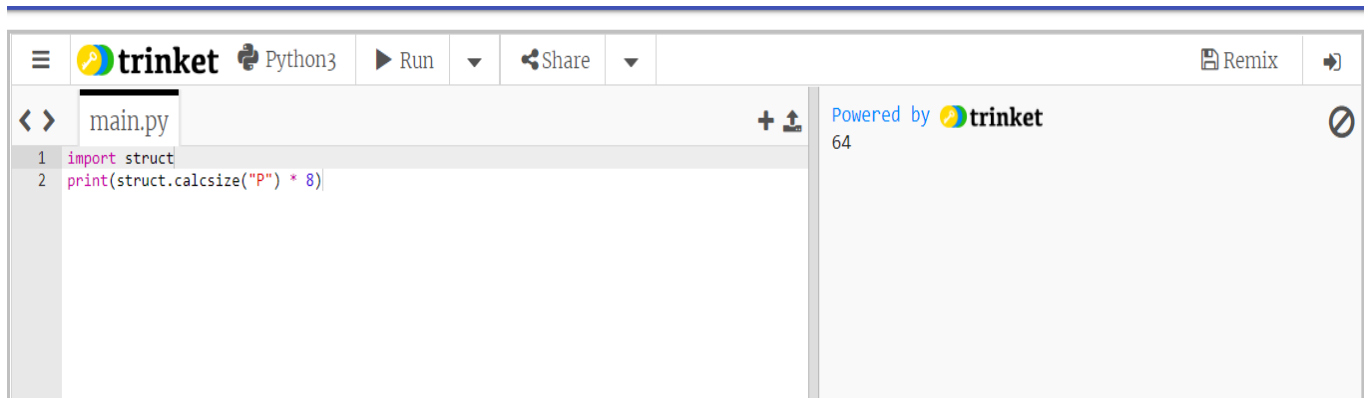
```
1e
04
```

#### Question-2:

Write a Python program to determine if the python shell is executing in 32bit or 64bit mode on operating system.

**Solution:**

```
import struct  
print(struct.calcsize("P") * 8)
```

**Output:**

The screenshot shows the Trinket Python IDE interface. The top bar includes the Trinket logo, a Python3 icon, and buttons for 'Run', 'Share', 'Remix', and a share icon. The main editor area displays a file named 'main.py' with the following code:

```
1 import struct  
2 print(struct.calcsize("P") * 8)
```

To the right of the code editor, the output area shows the result of the execution: '64'. The text 'Powered by trinket' is also visible in the output area.

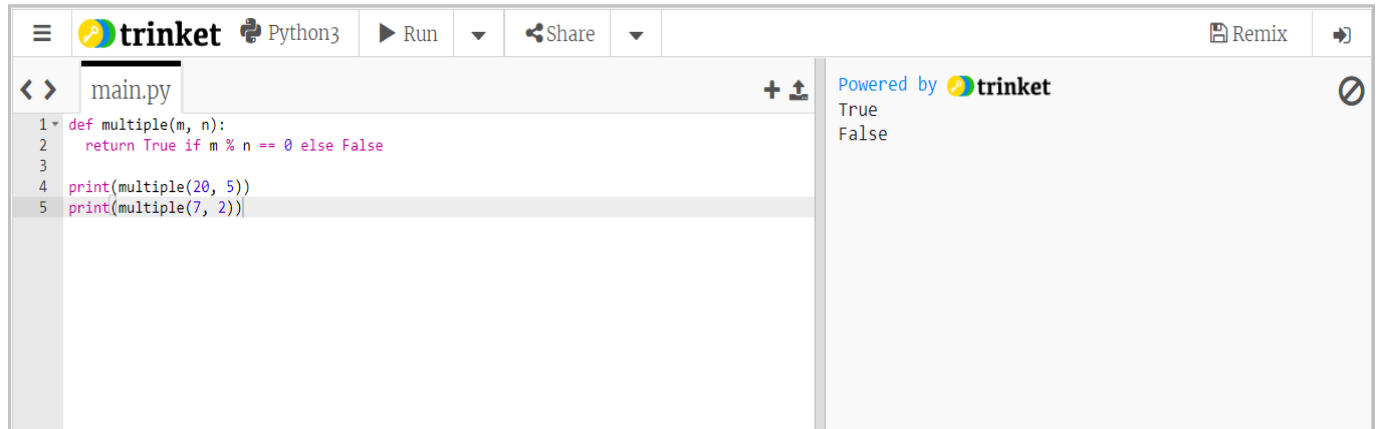
**Question-3:**

Write a Python function to check whether a number is divisible by another number. Accept two integers values from the user.

**Solution:**

```
def multiple(m, n):  
    return True if m % n == 0 else False  
print(multiple(20, 5))  
print(multiple(7, 2))
```

**Output:**



The screenshot shows a Trinket Python3 editor interface. The top bar includes the Trinket logo, 'Python3', 'Run', 'Share', 'Remix', and a share icon. The main editor area shows a file named 'main.py' with the following code:

```
1 def multiple(m, n):
2     return True if m % n == 0 else False
3
4 print(multiple(20, 5))
5 print(multiple(7, 2))
```

On the right side, there is a console area titled 'Powered by trinket' showing the output of the code:

```
True
False
```

#### Question-4:

Write a Python function to check whether a distinct pair of numbers whose product is odd present in a sequence of integer values

#### Solution:

```
def odd_product(nums):
    for i in range(len(nums)):
        for j in range(len(nums)):
            if i != j: product = nums[i] * nums[j]
            if product & 1:
                return True
    return False dt1 = [2, 4, 6, 8] dt2 = [1, 6, 4, 7, 8] dt3 = [1, 3, 5, 7, 9]
print(dt1, odd_product(dt1));
print(dt2, odd_product(dt2));
print(dt3, odd_product(dt3));
```

#### Output:

The screenshot shows a Trinket Python3 IDE interface. The editor on the left contains a Python script named 'main.py'. The script defines a function 'odd\_product(nums)' that iterates through pairs of elements in a list 'nums' and returns True if their product is odd. It then tests this function with three lists: [2, 4, 6, 8], [1, 6, 4, 7, 8], and [1, 3, 5, 7, 9]. The output on the right shows the results: False for the first list, True for the second, and True for the third.

```
1 def odd_product(nums):
2     for i in range(len(nums)):
3         for j in range(len(nums)):
4             if i != j:
5                 product = nums[i] * nums[j]
6                 if product & 1:
7                     return True
8     return False
9 dt1 = [2, 4, 6, 8]
10 dt2 = [1, 6, 4, 7, 8]
11 dt3 = [1, 3, 5, 7, 9]
12 print(dt1, odd_product(dt1));
13 print(dt2, odd_product(dt2));
14 print(dt3, odd_product(dt3));
```

Powered by trinket

```
[2, 4, 6, 8] False
[1, 6, 4, 7, 8] True
[1, 3, 5, 7, 9] True
```

### Question-5:

Write a Python function that takes a positive integer and returns the sum of the cube of all the positive integers smaller than the specified number.

### Solution:

```
def sum_of_cubes(n):
    n -= 1
    total = 0
    while n > 0:
        total += n * n * n
        n -= 1
    return total
print("Sum of cubes smaller than the specified number:",sum_of_cubes(3))
```

### Output:

trinket

Python3

Run

Share

Remix

<>

main.py

+ ↕

```
1 def sum_of_cubes(n):
2     n -= 1
3     total = 0
4     while n > 0:
5         total += n * n * n
6         n -= 1
7     return total
8 print("Sum of cubes smaller than the specified number: ",sum_of_cubes(3))
```

Powered by  trinket

Sum of cubes smaller than the specified number: 9