

ASSIGNMENT -3

Q1. Write a python program to generate the square of numbers from 1 to 10.

```
>>> from math import *
>>> for i in range(1,11):
...     i=(i*i)
...     print(i)
...
1
4
9
16
25
36
49
64
81
100
```

Q2. Write a function that test whether the given number is multiple of 5 or not.

```
>>> def f(x):
...     if x%5==0:
...         print(x,'is multiple of 5')
...     else:
...         print(x,'is not multiple of 5')
...
>>> f(10)
```

10 is multiple of 5

```
>>> f(12)
```

12 is not multiple of 5

Q3. Write a function that test whether the given number is divisible by 2,3, 5.

```
>>> def f(x):
...     if x%2==0:
...         if x%3==0:
```

```

...     if x%5==0:
...         print(x,'is divisible by 2,3 and 5')
...
>>> f(30)

```

30 is divisible by 2, 3 and 5

Q4. Generate all the prime numbers between 1 to 100 using python code.

```

>>> from math import *
>>> for i in range (2,101):
...     for j in range(2,101):
...         if i%j==0:
...             break
...         if i == j:
...             print(i,end=",")
...
2,3,5,7,11,13,17,19,23,29,31,37,41,43,47,53,59,61,67,71,73,79,83,89,97,>>>

```

Q5. Write a python program to generate the square of numbers from 20 to 30.

```

>>> from math import *
>>> for i in range(20,31):
...     i=(i*i)
...     print(i)
...

```

400

441

484

529

576

625

676

729

784

841

900

Q6. Write a function that test whether the given number is divisible by 2 or 3 or 5.

```
>>> def f(x):
...     if x%2==0:
...         print(x,'is divisible by 2')
...     else:
...         if x%3==0:
...             print(x,'is divisible by 3')
...         else:
...             if x%5==0:
...                 print(x,'is divisible by 5')
... 
```

```
>>> f(10)
10 is divisible by 2
```

```
>>> f(15)
15 is divisible by 3
```

```
>>> f(25)
25 is divisible by 5
```

Q7. Use python code to find the sum of first thirty natural numbers.

```
>>> sum=0
>>> n=1
>>> while n<=30:
...     sum=sum+n
...     n=n+1
...
>>> print('sum=',sum)
sum= 465
```

Q8. Write a python code to find the square of odd numbers from 1 to 20 using while loop.

```
>>> n=1
>>> while n<=20:
...     if n%2!=0:
...         square=n**2
...         print(f"The square of{n}is:{square}")
...         n+=1
... 
```

The square of 1 is:1

The square of 3 is:9

The square of 5 is:25

The square of 7 is:49

The square of 9 is:81

The square of 11 is:121

The square of 13 is:169

The square of 15 is:225

The square of 17 is:289

The square of 19 is:361

Q10. Use python code to generate the square root of numbers from 21 to 49.

```
>>> from math import *
```

```
>>> for n in range(21,50):
```

```
...     square_root=sqrt(n)
```

```
...     print(f"The square root of {n} is:{square_root}")
```

```
...
```

The square root of 21 is:4.58257569495584

The square root of 22 is:4.69041575982343

The square root of 23 is:4.795831523312719

The square root of 24 is:4.898979485566356

The square root of 25 is:5.0

The square root of 26 is:5.0990195135927845

The square root of 27 is:5.196152422706632

The square root of 28 is:5.291502622129181

The square root of 29 is:5.385164807134504

The square root of 30 is:5.477225575051661

The square root of 31 is:5.5677643628300215

The square root of 32 is:5.656854249492381

The square root of 33 is:5.744562646538029

The square root of 34 is:5.830951894845301

The square root of 35 is:5.916079783099616

The square root of 36 is:6.0

The square root of 37 is:6.082762530298219

The square root of 38 is: 6.164414002968976

The square root of 39 is: 6.244997998398398

The square root of 40 is: 6.324555320336759

The square root of 41 is: 6.4031242374328485

The square root of 42 is: 6.48074069840786

The square root of 43 is: 6.557438524302

The square root of 44 is: 6.6332495807108

The square root of 45 is: 6.708203932499369

The square root of 46 is: 6.782329983125268

The square root of 47 is: 6.855654600401044

The square root of 48 is: 6.928203230275509

The square root of 49 is: 7.0

Q11. Generate all relatively prime numbers to 111 which are less than 150 using python.

```
>>> from math import *
```

```
>>> def gcd(a,b):
```

```
...     while b:
```

```
...         a,b = b,a % b
```

```
...     return a
```

```
...
```

```
>>> def are_relatively_prime(num1,num2):
```

```
...     return gcd(num1,num2)==1
```

```
...
```

```
>>> base_number=111
```

```
>>> upper_limit=150
```

```
>>> relatively_prime_numbers=[num for num in range(1,upper_limit)if  
are_relatively_prime(base_number,num)]
```

```
>>> print(relatively_prime_numbers)
```

```
[1, 2, 4, 5, 7, 8, 10, 11, 13, 14, 16, 17, 19, 20, 22, 23, 25, 26, 28, 29, 31, 32, 34, 35, 38, 40, 41, 43, 44, 46, 47, 49,  
50, 52, 53, 55, 56, 58, 59, 61, 62, 64, 65, 67, 68, 70, 71, 73, 76, 77, 79, 80, 82, 83, 85, 86, 88, 89, 91, 92, 94, 95,  
97, 98, 100, 101, 103, 104, 106, 107, 109, 110, 112, 113, 115, 116, 118, 119, 121, 122, 124, 125, 127, 128, 130,  
131, 133, 134, 136, 137, 139, 140, 142, 143, 145, 146, 149]
```

Q12. Find numbers between 1 to 200 which are divisible by 7 using python code.

```
>>> n=[num for num in range(1,200)if num%7==0]
```

```
>>> print(n)
```

[7, 14, 21, 28, 35, 42, 49, 56, 63, 70, 77, 84, 91, 98, 105, 112, 119, 126, 133, 140, 147, 154, 161, 168, 175, 182, 189, 196]

Q13. Write python code to generate cube of numbers from 1 to 50.

```
>>> from math import *
```

```
>>> for i in range (1,51):
```

```
...     i=(i**3)
```

```
...     print(i)
```

```
...
```

```
1
```

```
8
```

```
27
```

```
64
```

```
125
```

```
216
```

```
343
```

```
512
```

```
729
```

```
1000
```

```
1331
```

```
1728
```

```
2197
```

```
2744
```

```
3375
```

```
4096
```

```
4913
```

```
5832
```

```
6859
```

```
8000
```

```
9261
```

```
10648
```

```
12167
```

```
13824
```

```
15625
```

```
17576
```

19683
21952
24389
27000
29791
32768
35937
39304
42875
46656
50653
54872
59319
64000
68921
74088
79507
85184
91125
97336
103823
110592
117649
125000

Q14. Write python code to generate 10 terms of Fibonacci sequence using loop.

```
>>> a=0
>>> b=1
>>> for i in range(10):
...     a,b=b,a+b
...     print(a,end=' ')
...
1 1 2 3 5 8 13 21 34 55 >>>
```

Q15. Use while command on python to find odd positive integers between 25 to 50.

```

>>> n=25

>>> while n<=50:
...     if n%2!=0:
...         print(n)
...     n +=1
...
25
27
29
31
33
35
37
39
41
43
45
47
49

```

Q16. Write python program to find the sum and product of n natural number using while loop.

```

>>> sum=0

>>> product=1

>>> n=1

>>> while n<=10:
...     sum=sum+n
...     product=product*n
...     n=n+1
...

>>> print('sum=',sum)

sum= 55

>>> print('product=',product)

product= 3628800

```

Q17. Write python code to find average of numbers 50 to 100.


```
>>> from math import *
>>> start_number=50
>>> end_number=100
>>> numbers = range(start_number,end_number + 1)
>>> average = sum(numbers)/len(numbers)
>>> print(f"the average of number from{start_number}to{end_number}is:{average}")
the average of number from50to100is:75.0
```

Q18. Use python loop to print('HELLO',i,'Do You Lesrn Python') where i=['Saurabh','Akash','Sandeep','Ram','Sai'].

```
>>> for i in ['Saurabh','Akash','Sandeep','Ram','Sai']:
```

```
...     print('Hello' , i , 'Do You Learn Python')
```

```
...
```

```
Hello Saurabh Do You Learn Python
```

```
Hello Akash Do You Learn Python
```

```
Hello Sandeep Do You Learn Python
```

```
Hello Ram Do You Learn Python
```

```
Hello Sai Do You Learn Python
```

Q19. Using for loop on python, find range from 1 to 11 integers.

```
>>> for i in range(1,12):
```

```
...     print(i)
```

```
...
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

```
10
```

```
11
```

Q20. Write python code to display multiplication tables of numbers 2 to 10.

```
>>> for i in range(2,11):
```

```
... for j in range(1,11):  
...     print(f"{i}x{j}={i*j}")
```

```
...
```

2x1=2

2x2=4

2x3=6

2x4=8

2x5=10

2x6=12

2x7=14

2x8=16

2x9=18

2x10=20

3x1=3

3x2=6

3x3=9

3x4=12

3x5=15

3x6=18

3x7=21

3x8=24

3x9=27

3x10=30

4x1=4

4x2=8

4x3=12

4x4=16

4x5=20

4x6=24

4x7=28

4x8=32

4x9=36

4x10=40

$$5 \times 1 = 5$$

$$5 \times 2 = 10$$

$$5 \times 3 = 15$$

$$5 \times 4 = 20$$

$$5 \times 5 = 25$$

$$5 \times 6 = 30$$

$$5 \times 7 = 35$$

$$5 \times 8 = 40$$

$$5 \times 9 = 45$$

$$5 \times 10 = 50$$

$$6 \times 1 = 6$$

$$6 \times 2 = 12$$

$$6 \times 3 = 18$$

$$6 \times 4 = 24$$

$$6 \times 5 = 30$$

$$6 \times 6 = 36$$

$$6 \times 7 = 42$$

$$6 \times 8 = 48$$

$$6 \times 9 = 54$$

$$6 \times 10 = 60$$

$$7 \times 1 = 7$$

$$7 \times 2 = 14$$

$$7 \times 3 = 21$$

$$7 \times 4 = 28$$

$$7 \times 5 = 35$$

$$7 \times 6 = 42$$

$$7 \times 7 = 49$$

$$7 \times 8 = 56$$

$$7 \times 9 = 63$$

$$7 \times 10 = 70$$

$$8 \times 1 = 8$$

$$8 \times 2 = 16$$

$$8 \times 3 = 24$$

$$8 \times 4 = 32$$

$$8 \times 5 = 40$$

$$8 \times 6 = 48$$

$$8 \times 7 = 56$$

$$8 \times 8 = 64$$

$$8 \times 9 = 72$$

$$8 \times 10 = 80$$

$$9 \times 1 = 9$$

$$9 \times 2 = 18$$

$$9 \times 3 = 27$$

$$9 \times 4 = 36$$

$$9 \times 5 = 45$$

$$9 \times 6 = 54$$

$$9 \times 7 = 63$$

$$9 \times 8 = 72$$

$$9 \times 9 = 81$$

$$9 \times 10 = 90$$

$$10 \times 1 = 10$$

$$10 \times 2 = 20$$

$$10 \times 3 = 30$$

$$10 \times 4 = 40$$

$$10 \times 5 = 50$$

$$10 \times 6 = 60$$

$$10 \times 7 = 70$$

$$10 \times 8 = 80$$

$$10 \times 9 = 90$$

$$10 \times 10 = 100$$

Q21. Write python code to check if a number is Zero,Odd or Even.

```
>>> def f(x):
```

```
...     if x<0:
```

```
...         print(x,'is negetive')
```

```
...     else:
```

```
...         if x>0:
```

```

...     print(x,'is positive')
...     else:
...         print('Given number is zero')
...

```

```

>>> f(15)

15 is positive

```

```

>>> f(-5)

-5 is negetive

```

```

>>> f(0)

Given number is zero

```

Q22. Write python code to print first n natural number and their square roots of input integer n.

```

>>> from math import *

>>> n=int(input("Enter a positive integer(n):"))

Enter a positive integer(n):10

>>> for i in range(1,n+1):
...     square_root=sqrt(i)
...     print(f"Number:{i},Square Root:{square_root}")
...

```

```

Number:1,Square Root:1.0
Number:2,Square Root:1.4142135623730951
Number:3,Square Root:1.7320508075688772
Number:4,Square Root:2.0
Number:5,Square Root:2.23606797749979
Number:6,Square Root:2.449489742783178
Number:7,Square Root:2.6457513110645907
Number:8,Square Root:2.8284271247461903
Number:9,Square Root:3.0
Number:10,Square Root:3.1622776601683795

```

Q23. Use python code to find sum of square of first 25 natural number >>> from math import *

```

>>> sum_of_squares=sum(i**2 for i in range(1,26))

>>> print(f"The sum of the squares of the first 25 natural numbers is:{sum_of_squares}")

```

The sum of the squares of the first 25 natural numbers is:5525

Q24. Write python code to find all the positive divisor of given number n.

```
>>> def divisor(x):
...     for i in range(1,x+1):
...         if x%i==0:
...             print(i,end=' ')
...
>>> divisor(50)
1 2 5 10 25 50 >>>
```

Q25. Write python code to prints all integers between 1 to 100 that are divisible by 3 and 7.

```
>>> for i in range(1,101):
...     if i%3==0 and i%7==0:
...         print(i,end=' ')
...
21 42 63 84 >>>
```

Q27. . Write python program to prints whether the given number is divisible by 3 or 5 or 7

```
>>> def f(x):
...     if x%3==0:
...         print(x,'is divisible by 3')
...     else:
...         if x%5==0:
...             print(x,'is divisible by 5')
...         else:
...             if x%7==0:
...                 print(x,'is divisible by 7')
...
>>> f(9)
9 is divisible by 3
>>> f(15)
15 is divisible by 5
>>> f(28)
28 is divisible by 7
```

Q28. Use python code to find the remainder after dividing by 'n' to any integers.

```
>>> def remainder(integer,n):
...     remainder=integer % n
...     return remainder
...
>>> num=17
>>> divisor=5
>>> result=remainder(num,divisor)
>>> print(f"The remainder after dividing{num}by{divisor}is:{result}")
```

The remainder after dividing17by5is:2

Q29. Write python program to find perfect square between 1 to 100.

```
>>> def perfect_square(start,end):
...     perfect_square=[i**2 for i in range(int(start**0.5),int(end**0.5)+1)]
...     return perfect_square
...
>>> start_range=1
>>> end_range=100
>>> perfect_square_list=perfect_square(start_range,end_range)
>>> print(f"Perfect squares between {start_range}and{end_range}:{perfect_square_list}")

Perfect squares between 1and100:[1, 4, 9, 16, 25, 36, 49, 64, 81, 100]
>>>
```

Q30. Write python program to prints whether the given natural number is divisible by 5 and less than 100.

```
>>> def divisibility(n):
...     if n>0 and n%5==0 and n<100:
...         print(n,'is divisible 5 and less than 100')
...     else:
...         print(n,'is either not divisible by 5 or not less than 100')
...
>>> divisibility(10)

10 is divisible 5 and less than 100
>>> divisibility(115)

115 is either not divisible by 5 or not less than 100
```

