CYCLE-1

1. Program to Print all non-Prime Numbers in an Interval

CODE

```
import math
def is_not_prime(n):
    ans = False
    for i in range(2, int(math.sqrt(n)) + 1):
        if n % i == 0:
            ans = True
    return ans
print("Nonprime numbers between 1 to 100:")
for x in filter(is_not_prime, range(1, 101)):
    print(x)
```

OUTPUT

```
Nonprime numbers between 1 to 100:
4
6
8
9
10
12
14
15
16
18
```

2. Program to print the first N Fibonacci numbers.

Code

```
n = int(input("How many terms? "))
a, b = 0, 1
count = 0
if n <= 0:
  print("Please enter a positive integer")

elif n == 1:
  print("Fibonacci sequence upto",n,":")
  print(a)
else:
  print("Fibonacci sequence:")
  while count < n:
    print(a)
    c = a + b
    a = b
    b = c
    count += 1
```

OUTPUT

```
How many terms? 5
Fibonacci sequence:
0
1
1
2
3

Process finished with exit code 0
```

3. Given sides of a triangle, write a program to check whether given triangle is an isosceles, equilateral or scalene.

Code

```
s1=float(input("Enter the side 1 of the triangle\n"))
s2=float(input("Enter the side 2 of the triangle\n"))
s3=float(input("Enter the side 3 of the triangle\n"))
if(s1==s2 and s2==s3):
    print("Equilateral Triangle")
elif(s1==s2 or s2==s3 or s3==s1):
    print("Isosceles triangle")
else:
    print("scalen Triangle")
```

OUTPUT

```
Enter the side 1 of the triangle
2
Enter the side 2 of the triangle
2
Enter the side 3 of the triangle
5
Isosceles triangle

Process finished with exit code 0
```

4. Program to check whether given pair of number is coprime.

Code

```
a=int(input("Enter no:1 \n"))
b=int(input("Enter no:2 \n"))
for i in range (1,a):
    if(a%i==0 and b%i==0):
```

```
        hcf=i
if(hcf==1):
    print("coprime")
else:
    print("not coprime")
```

Output

```
Enter no:1
2
Enter no:2
5
coprime

Process finished with exit code 0
```

5. Program to find the roots of a quadratic equation(rounded to 2 decimal places)

Code

```
import cmath
a=float(input("Enter the value of a \n"))
b=float(input("Enter the value of b \n"))
c=float(input("Enter the value of c \n"))
d=(b*b)-(4*a*c)
if(d>0):
    r1= (-b + cmath.sqrt(d)) / (2 * a)
    r2 =(-b - cmath.sqrt(d)) / (2 * a)
    print("The roots are real and different",r1,r2)
elif(d==0):
```

```
    r1=r2=-b/2*a
    print("roots are real and  equal",r1)
else:
   real = -b / (2 * a)
   img = cmath.sqrt(d) / (2 * a)
   print("complex roots",real,"+",img, "and",real,"-",img)
```

OUTPUT

```
Enter the value of a
1
Enter the value of b
7
Enter the value of c
12
The roots are real and different (-3+0j) (-4+0j)

Process finished with exit code 0
```

6. Program to check whether a given number is perfect number or not(sum of factors =number)

CODE

```
n = int(input("Enter any number: "))
sum = 0
for i in range(1, n):
    if(n % i == 0):
        sum = sum + i
if (sum == n):
    print("The number is a Perfect number")
else:
    print("The number is not a Perfect number")
```

output

```
Enter any number: 25
The number is not a Perfect number

Process finished with exit code 0
```

7. Program to display amstrong numbers upto 1000.

Code

```
lower = 100
upper = 1000

for num in range(lower, upper + 1):


    order = len(str(num))


    sum = 0

    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** order
        temp //= 10

    if num == sum:
        print(num)
```

Output

```
153
370
371
407

Process finished with exit code 0
```

8. Store and display the days of a week as a List, Tuple, Dictionary, Set. Also demonstrate different ways to store values in each of them. Display its type also.

Code

```
list = ["Sun","Mon","Tue","Wed","Thu","Fri","Sat"]

print(type(list))

print(list)

tuple = ("Sun","Mon","Tue","Wed","Thu","Fri","Sat")

print(type(tuple))

print(tuple)

set = {"Sun","Mon","Tue","Wed","Thu","Fri","Sat"}

print(type(set))

print(set)

dict = {

    "d1" : "Sun",

    "d2" : "Mon",

    "d3" : "Tue",
```

```
    "d4" : "Wed",
    "d5" : "Thu",
    "d6" : "Fri",
    "d7" : "Sat"
}
print(type(dict))
print(dict)
```

output

```
<class 'list'>
['Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat']
<class 'tuple'>
('Sun', 'Mon', 'Tue', 'Wed', 'Thu', 'Fri', 'Sat')
<class 'set'>
{'Wed', 'Sat', 'Fri', 'Sun', 'Thu', 'Tue', 'Mon'}
<class 'dict'>
{'d1': 'Sun', 'd2': 'Mon', 'd3': 'Tue', 'd4': 'Wed', 'd5': 'Thu', 'd6': 'Fri', 'd7': 'Sat'}

Process finished with exit code 0
```

9. Write a program to add elements of given 2 lists.

Code

```
List1 = [10, 20, 30, 45]
List2 = [15, 25, 35, 56]
total = []

for j in range(4):
    total.append(List1[j] + List2[j])

print("\nThe total Sum of Two Lists =  ", total)
```

output

```
The total Sum of Two Lists =   [25, 45, 65, 101]

Process finished with exit code 0
```

10. Write a program to find the sum of 2 matrices using nested List.

Code

```
X = [[12,7,3],
   [4 ,5,6],
   [7 ,8,9]]

Y = [[5,8,1],
   [6,7,3],
   [4,5,9]]

result = [[X[i][j] + Y[i][j]
         for j in range(len(X[0]))]
        for i in range(len(X))]

for r in result:
  print(r)
```

output

```
[17, 15, 4]
[10, 12, 9]
[11, 13, 18]

Process finished with exit code 0
```

11. Write a program to perform bubble sort on a given set of elements.

Code

```
a = [35, 10, 31, 11, 26]
```

```
print("Before sorting array elements are - ")
for i in a:
   print(i, end = " ")
for i in range(0,len(a)):
   for j in range(i+1,len(a)):
      if a[j]<a[i]:
         temp = a[j]
         a[j]=a[i]
         a[i]=temp
print("\nAfter sorting array elements are - ")
for i in a:
   print(i, end = " ")
```

output

```
Before sorting array elements are -
35 10 31 11 26
After sorting array elements are -
10 11 26 31 35
Process finished with exit code 0
```

12. Program to find the count of each vowel in a string(use dictionary).

Code

```
string=input("Enter string:")
vowels=0
for i in string:
    if(i=='a' or i=='e' or i=='i' or i=='o' or i=='u' or i=='A' or i=='E' or i=='I' or
i=='O' or i=='U'):
        vowels=vowels+1
print("Number of vowels are:")
```

```
print(vowels)
```

Output

```
Enter string:umberlla
Number of vowels are:
3

Process finished with exit code 0
```

13. Write a Python program that accept a positive number and subtract from this number the sum of its digits and so on. Continues this operation until the number is positive.

Code

```
def repeat_times(n):
  s = 0
  n_str = str(n)
  while (n > 0):
    n -= sum([int(i) for i in list(n_str)])
    n_str = list(str(n))
    s += 1
  return s
print(repeat_times(9))
```

```
print(repeat_times(21))
```

output

```
1
3

Process finished with exit code 0
```

14. Write a Python program that accepts a 10 digit mobile number, and find the digits which are absent in a given mobile number.

Code

```
def absent_digits(n):
  all_nums = set([0,1,2,3,4,5,6,7,8,9])
  n = set([int(i) for i in n])
  n = n.symmetric_difference(all_nums)
  n = sorted(n)
  return n
print(absent_digits([9,8,3,2,2,0,9,7,6,3]))
```

output

```
[1, 4, 5]

Process finished with exit code 0
```