DATA SCIENCE LAB

CYCLE1

SUBMITTED BY,

AMAL T S
20MCA011

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

Code

```
lower = int(input("Enter lower range: "))
upper = int(input("Enter the upper range: "))
for num in range(lower, upper):
   if num > 1:
      for i in range(2, num):
         if(num % i) == 0:
            print(num)
            break
```

output

```
Enter lower range: 5
Enter the upper range: 12
6
8
9
10

Process finished with exit code 0
```

2. **Program to print the first N Fibonacci numbers.**

Code

```
n = int(input("enter n terms"))
if n < 0:
   print("Enter a positive number")
```

```
else:
    f1 = 0
    f2 = 1
    if n == 1:
        print(f1)
    elif n == 2:
        print(f1)
        print(f2)
    else:
        print(f1)
        print(f2)
        for i in range(3, n):
            f3 = f1 + f2
            print(f3)
            f1 = f2
            f2 = f3
```

output

```
enter n terms7
0
1
1
2
3
5

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

```
print("Input lengths of the triangle sides: ")
x = int(input("x: "))
y = int(input("y: "))
z = int(input("z: "))
if x == y == z:
    print("It is a Equilateral triangle")
elif x == y or y == z or z == x:
    print("It is a Isosceles triangle")
else:
    print("It is a Scalene triangle")
```

output

```
Input lengths of the triangle sides:
x: 4
y: 5
z: 6
It is a Scalene triangle

Process finished with exit code 0
```

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

Code

```
def gcd(p, q):
```

```
    while q != 0:
        p, q = q, p % q
    return p
def is_coprime(x, y):
    return gcd(x, y) == 1
print (is_coprime(17, 13))
print (is_coprime(5, 7))
print (is_coprime(10, 20))
```

Output

```
True
True
False

Process finished with exit code 0
```

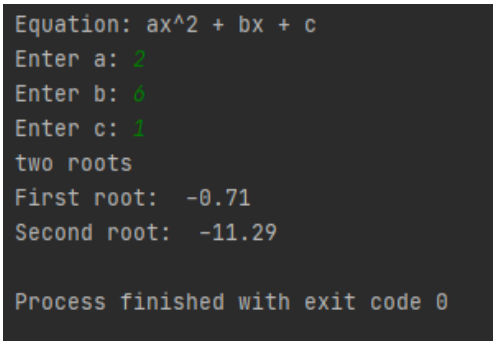
5. **Program to find the roots of a quadratic equation(rounded to 2 decimal places)**

Code

```
print("Equation: ax^2 + bx + c")
a = int(input("Enter a: "))
b = int(input("Enter b: "))
c = int(input("Enter c: "))
d = b ** 2 - 4 * a * c
d1 = d**0.5
```

```
if d < 0:
    print("The roots are imaginary.")
elif d == 0:
    r = -b / 2 * a
    print("single root: ", round(r, 2))
else:
    r1 = (-b + d1)/2 * a
    r2 = (-b - d1)/2 * a
    print("two roots")
    print("First root: ", round(r1, 2))
    print("Second root: ", round(r2, 2))
```

output



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

Code

```
a = int(input("enter the number: "))
sum = 0
```

```
for i in range(1, a):
    if a % i == 0:
        sum = sum + i
if sum == a:
    print("number is perfect number")
else:
    print("number is not a perfect number")
```

output

```
enter the number: 6
number is perfect number

Process finished with exit code 0
```

7. **Program to display amstrong numbers upto 1000**

Code

```
lower = int(input("Enter lower range: "))
upper = int(input("Enter upper range: "))
for num in range(lower, upper + 1):
    sum = 0
    temp = num
    while temp > 0:
        digit = temp % 10
        sum += digit ** 3
        temp //= 10
```

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

output

```
Enter lower range: 100
Enter upper range: 500
125
153
216
370
371
407
```

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'>
{'Sun', 'Mon', 'Tue', 'Fri', 'Wed', 'Sat', 'Thu'}
<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

```
l1 = [1, 2, 3, 4]

l2 = [4, 9, 0, 1]

l1.extend(l2)
print(l1)
```

output

```
[1, 2, 3, 4, 4, 9, 0, 1]

Process finished with exit code 0
```

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

Code

```
matOne = []
print("Enter 9 Elements for First Matrix: ")
for i in range(3):
    matOne.append([])
    for j in range(3):
        num = int(input())
        matOne[i].append(num)

matTwo = []
print("Enter 9 Elements for Second Matrix: ")
for i in range(3):
    matTwo.append([])
    for j in range(3):
        num = int(input())
        matTwo[i].append(num)

matThree = []
for i in range(3):
    matThree.append([])
```

```
    for j in range(3):
        matThree[i].append(matOne[i][j]+matTwo[i][j])


print("\nAddition Result of Two Given Matrix is:")
for i in range(3):
    for j in range(3):
        print(matThree[i][j], end=" ")
    print()
```

output

```
Enter 9 Elements for First Matrix:
4
5
5
6
3
2
2
5
6
Enter 9 Elements for Second Matrix:
8
5
54
2
5
6
5
5
8

Addition Result of Two Given Matrix is:
12 10 59
8 8 8
7 10 14

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 11 26 31 35
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:amal
Number of vowels are:
2
```

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
n = int(input("enter number"))
p = int(input("enter number"))
print(repeat_times(n))
print(repeat_times(p))
```

output

```
enter number9
enter number22
1
3
```

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 are: ", absent_digits([9,5,2,6,0,1,4,6,8,4])
```

output

```
absent digits are:  [3, 7]

Process finished with exit code 0
```