



Python Programming - 2301CS404

Lab - 4

Roll No.: 369

Name: Kalola Hill

01) WAP to print 1 to 10.

```
In [1]: print("Print 1 to 10:")
        for i in range(1, 11):
            print(i, end=" ")
        print()

Print 1 to 10:
1 2 3 4 5 6 7 8 9 10
```

02) WAP to print 1 to n.

```
In [2]: print("Print 1 to n:")
        n = int(input("Enter value of n: "))
        print(f"Numbers from 1 to {n}:")
        for i in range(1, n + 1):
            print(i, end=" ")
        print()

Print 1 to n:
Numbers from 1 to 6:
1 2 3 4 5 6
```

03) WAP to print odd numbers between 1 to n.

```
In [3]: print("Print odd numbers between 1 to n:")
        n = int(input("Enter value of n: "))
        print(f"Odd numbers from 1 to {n}:")
        for i in range(1, n + 1):
            if i % 2 != 0:
                print(i, end=" ")
        print()

Print odd numbers between 1 to n:
Odd numbers from 1 to 9:
1 3 5 7 9
```

04) WAP to print numbers between two given numbers which is divisible by 2 but not divisible by 3.

```
In [4]: print("Numbers divisible by 2 but not by 3:")
        start = int(input("Enter starting number: "))
        end = int(input("Enter ending number: "))

        print(f"Numbers between {start} and {end} divisible by 2 but not by 3:")
        for num in range(start, end + 1):
            if num % 2 == 0 and num % 3 != 0:
                print(num, end=" ")
        print()

Numbers divisible by 2 but not by 3:
Numbers between 3 and 12 divisible by 2 but not by 3:
4 8 10
```

05) WAP to print sum of 1 to n numbers.

```
In [5]: print("Sum of 1 to n numbers:")
        n = int(input("Enter value of n: "))
        total = 0
        for i in range(1, n + 1):
            total += i
        print(f"Sum of numbers from 1 to {n} = {total}")

Sum of 1 to n numbers:
Sum of numbers from 1 to 9 = 45
```

06) WAP to print sum of series 1 + 4 + 9 + 16 + 25 + 36 + ...n.

```
In [6]: print("Sum of squares series:")
        n = int(input("Enter value of n: "))
        sum_squares = 0
        for i in range(1, n + 1):
            sum_squares += i ** 2
        print(f"Sum of series 1^2 + 2^2 + ... + {n}^2 = {sum_squares}")

Sum of squares series:
Sum of series 1^2 + 2^2 + ... + 7^2 = 140
```

07) WAP to print sum of series 1 – 2 + 3 – 4 + 5 – 6 + 7 ... n.

```
In [7]: print("Sum of alternating series:")
        n = int(input("Enter value of n: "))
        alt_sum = 0
        for i in range(1, n + 1):
            if i % 2 == 0:
                alt_sum -= i
            else:
                alt_sum += i
        print(f"Sum of series 1-2+3-4+...+{n} = {alt_sum}")

Sum of alternating series:
Sum of series 1-2+3-4+...+8 = -4
```

08) WAP to print Multiplication Table of the given number.

```
In [8]: print("Multiplication Table:")
        num = int(input("Enter a number: "))
```

```
print(f"Multiplication table of {num}:")
for i in range(1, 11):
    print(f"{num} x {i} = {num * i}")
```

Multiplication Table:  
Multiplication table of 6:  
6 x 1 = 6  
6 x 2 = 12  
6 x 3 = 18  
6 x 4 = 24  
6 x 5 = 30  
6 x 6 = 36  
6 x 7 = 42  
6 x 8 = 48  
6 x 9 = 54  
6 x 10 = 60

09) WAP to find Factorial of the given number.

```
In [9]: print("Factorial of a number:")
num = int(input("Enter a number: "))
factorial = 1
for i in range(1, num + 1):
    factorial *= i
print(f"Factorial of {num} = {factorial}")
```

Factorial of a number:  
Factorial of 7 = 5040

10) WAP to print GCD of given two numbers.

```
In [11]: print("GCD of two numbers:")
num1 = int(input("Enter first number: "))
num2 = int(input("Enter second number: "))

a, b = num1, num2
while b != 0:
    a, b = b, a % b
print(f"GCD of {num1} and {num2} = {a}")
```

GCD of two numbers:  
GCD of 6 and 18 = 6

11) WAP to find Factors of the given number.

```
In [12]: print("Factors of a number:")
num = int(input("Enter a number: "))
print(f"Factors of {num}:")
for i in range(1, num + 1):
    if num % i == 0:
        print(i, end=" ")
print()
```

Factors of a number:  
Factors of 6:  
1 2 3 6

12) WAP to find whether the given number is Prime or not.

```
In [13]: print("Prime number check:")
num = int(input("Enter a number: "))
is_prime = True

if num <= 1:
    is_prime = False
else:
    for i in range(2, int(num ** 0.5) + 1):
        if num % i == 0:
            is_prime = False
            break

if is_prime:
    print(f"{num} is a prime number")
else:
    print(f"{num} is not a prime number")
```

Prime number check:  
13 is a prime number

13) WAP to print sum of digits of given number.

```
In [14]: print("Sum of digits:")
num = int(input("Enter a number: "))
temp = abs(num)
digit_sum = 0

while temp > 0:
    digit = temp % 10
    digit_sum += digit
    temp //= 10

print(f"Sum of digits of {num} = {digit_sum}")
```

Sum of digits:  
Sum of digits of 1296 = 18

14) WAP to check whether the given number is Palindrome or not.

```
In [15]: print("Palindrome check:")
num = int(input("Enter a number: "))
temp = num
reverse = 0

while temp > 0:
    digit = temp % 10
    reverse = reverse * 10 + digit
    temp //= 10

if num == reverse:
    print(f"{num} is a palindrome")
else:
    print(f"{num} is not a palindrome")
```

Palindrome check:  
12321 is a palindrome

15) WAP to check whether the given number is an Armstrong Number or not.

```
In [16]: print("Armstrong number check:")
num = int(input("Enter a number: "))
temp = num
power = len(str(num))
```

```
armstrong_sum = 0

while temp > 0:
    digit = temp % 10
    armstrong_sum += digit ** power
    temp //= 10

if num == armstrong_sum:
    print(f"{num} is an Armstrong number")
else:
    print(f"{num} is not an Armstrong number")
```

Armstrong number check:  
153 is an Armstrong number

16) WAP to print all the perfect numbers between 1 to n.

```
In [18]: print("Perfect numbers between 1 to n:")
n = int(input("Enter value of n: "))
print(f"Perfect numbers between 1 and {n}:")

for num in range(1, n + 1):
    divisors_sum = 0
    for i in range(1, num):
        if num % i == 0:
            divisors_sum += i

    if divisors_sum == num:
        print(num, end=" ")

print("\n" + "="*50)
```

Perfect numbers between 1 to n:  
Perfect numbers between 1 and 184:  
6 28  
=====