Assignment 1

1 .Check the given number is odd or even

In [4]:

```
x = 24
# Check the remainder dividing x by 2 is 0
if x % 2 == 0:
 print ("Even")
else :
 print("odd")

# Checking another number
x = 7

if x % 2 == 0:

   print("Even")
else:
 print("odd")
```

```
Even
odd
```

1. Count the total number of digits in a number

In [1]:

```
def count_digits(number):
    """Coount the total number of digits in a number."""
    return len(str(number))
#Example usage
number = 12345
digit_count = count_digits(number)
print(f"The number of digits in {number} is: {digit_count}")
```

```
The number of digits in 12345 is: 5
```

1. Write a python program to print reverse number pattern.

In [3]:

```
def reverse_number_pattern(row):
    for i in range(row, 0, -1):
        for j in range(i, 0, -1):
            print(j, end=" ")
        print()
reverse_number_pattern(5)
```

```
5 4 3 2 1
4 3 2 1
3 2 1
```

```
2 1
1
```

1. print all prime number within a range

In [6]:

```
def print_primes_in_range(start,end):
    """Prints all prime number within the range [start, end] (inclusive)"""

    def is_prime(num):
        """Checks if a number is prime."""
        if num <= 1:
            return False
        for i in range(2, int(num**0.5) + 1):
            if num % i == 0:
                return False
        return True
    for num in range(start, end + 1):
       if is_prime(num):
           print(num)

#Example usage:
print_primes_in_range(2,50)
```

```
2
3
5
7
11
13
17
19
23
29
31
37
41
43
47
```

1. Find the factorial of a given number

In [6]:

```
def factorial(n):
    if n < 0:
        return "Factorial is not defined in negative numbers"
    elif n == 0:
        return 1
    else:
        fact = 1
        for i in range(1, n + 1):
            fact = fact * i
        return fact

number = int(input("Enter a number : "))
result = factorial(number)
print("The factorial of",number, "is", result)
```

```
Enter a number :  9

The factorial of 9 is 362880
```

1. Program to check if number palindrome

In [9]:

```
import math

def rev(num):
    return int(num != 0) and ((num % 10) * \
            (10**int(math.log(num, 10))) + \
                        rev(num // 10))
test_number = 1669669
print ("The original number is : " + str(test_number))

res = test_number == rev(test_number)
print ("Is the number palindrome ? : " + str(res))
```

```
The original number is : 1669669
Is the number palindrome ? : False
```

1. program to check Armstrong number

In [10]:

```
num=int(input("Enter a number:"))
order=len(str(num))
sum=0
temp=num
while temp >0:
    digit=temp%10
    sum+=digit**order
    temp//=10
    if num==sum:
        print(f"{num} is an armstrong number.")
    else:
        print(f"{num} is not an armstrong number.")
```

```
Enter a number: 9474

9474 is not an armstrong number.
9474 is not an armstrong number.
9474 is not an armstrong number.
9474 is an armstrong number.
```

8.Find maximum of three number

In [11]:

```
a=float(input("Enter First number:"))
b=float(input("Enter second number:"))
c=float(input("Enter third number:"))
maximum=max(a,b,c)
print(f"The maximum number is :{maximum}")
```

```
Enter First number: 23
Enter second number: 43
Enter third number: 56
```

```
The maximum number is :56.0
```

9.Find the sum of digits

In [26]:

```python
num=int(input("Enter a number :"))
num=abs(num)
sum_digits=0
while num >0:
    sum_digits +=num%10
    num//=10
    print("sum of digits:",sum_digits)
```

```
Enter a number : 1234

sum of digits: 4
sum of digits: 7
sum of digits: 9
sum of digits: 10
```

10.Python programs to print the Natural Numbers Summation Pattern Given natural number n, the task is to write a Python program to first find the sum of first n natural numbers and then print each step as a pattern.

In [2]:

```python
def print_summation_pattern(n):
    """
    Calculates the sum of natural numbers up to n and prints the summation

    Args:
        n (int): The upper limit of natural numbers to sum.
    """
    total_sum = 0
    for i in range(1, n + 1):
        total_sum += i
        expression = " + ".join(str(j) for j in range(1, i + 1))
        print(f"{expression} = {total_sum}")

# Example usage:
num = int(input("Enter a natural number: "))
print_summation_pattern(num)
```

```
Enter a natural number:  5

1 = 1
1 + 2 = 3
1 + 2 + 3 = 6
1 + 2 + 3 + 4 = 10
1 + 2 + 3 + 4 + 5 = 15
```