In [1]:
```python
#q1 calculate factorial

def calc_factorial(num):
    product = 1
    while num > 0:
        product = product * num
        num = num - 1
    return product

n = int(input("Enter a number: "))
print(calc_factorial(n))
```

```
120
```

In [28]:
```python
#q2 check prime

def check_prime(num):
    if num < 2:
        return False

    fact_count = 0
    for i in range(1, num // 2 + 1):
        if num % i == 0:
            fact_count += 1
    return fact_count == 1

print(check_prime(2))
print(check_prime(4))
print(check_prime(7))
print(check_prime(13))
print(check_prime(26))
```

```
True
False
True
True
False
```

In [7]:
```python
#q3 reverse a number

def reverse_num(num):
    dup = num
    rev = 0
    while dup > 0:
        rev = rev*10 + dup%10
        dup = dup // 10
    return rev
n = int(input("Enter a number to reverse: "))
print(reverse_num(n))
```

```
321
```

In [13]:
```python
#q4 return unique elements from a list

def find_unique(my_list){
    return set(my_list)
```

```
}

my_list = [1,2,3,4,5,5,5,6, "A", "A", "B"]
print(set()
```

```
{1, 2, 3, 4, 5, 6, 'A', 'B'}
```

In [30]:
```
#q5 count uppercase and lowercase letters

def count_upper_lower(string):
    upper_count = 0
    lower_count = 0
    for ch in string:
        if ch.isupper():
            upper_count += 1
        elif ch.islower():
            lower_count += 1

    return f'Upper: {upper_count}, Lower: {lower_count}'

print(count_upper_lower("My name is UTKARSH!"))
```

```
Upper: 8, Lower: 7
```

In [22]:
```
#q6 compute nCr

def find_fact(num):
    fact = 1
    for i in range(1, num+1):
        fact *= i
    return fact

def ncr(n, r):
    n_fact = find_fact(n)
    n_r_fact = find_fact(n-r)
    r_fact = find_fact(r)
    res = n_fact / (n_r_fact * r_fact)
    return res


print(ncr(5, 3))
```

```
10.0
```

In [24]:
```
#q7 check for perfect number

def check_perfect(num):
    sum = 0
    for i in range(1, num):
        if num%i == 0:
            sum += i
    return sum == num

print(check_perfect(6))
```

```
True
```

In [11]:
```python
#q8 compute square root without library

def find_square_root(num):
    x = num
    for i in range(20):
        x = 0.5*(x+num/x)
    return x

n = int(input("Enter a number: "))
print(find_square_root(n))
```

```
6.0
```

In [15]:
```python
#q9 lambda function to find cube of a number

def find_cube(num):
    cube = lambda x : x**3
    return(cube(num))

n = int(input("Enter a number: "))
print(find_cube(n))
```

```
27
```

In [20]:
```python
#q10 lambda function to extract even numbers from  a list

def extract_even(my_list):
    even_list = list(filter(lambda x : x%2 == 0, my_list))
    return even_list

print(extract_even([1,2,3,4,5,6,7,8]))
```

```
[2, 4, 6, 8]
```