Assignment – 2

Name – Alisha Khatoon
Mail_id : khatoonalisha275@gmail.com

1.Find the factorial of a given number
2. Reverse a integer number
3. Print elements from a given list present at odd index positions
4. Calculate the cube of all numbers from 1 to a given number
5. Find the sum of the series up to n terms
6. Get Smallest Number in List
7. Remove Duplicates from List
8. Remove Even Numbers from List
9. Find the Second Largest Number in List
10. Check If All Numbers Are Prime
11. Count Frequency of List Elements
12. Find the Longest Consecutive Sequence in a List
13. Rotate a List by K Positions
14. Find the Intersection of Two Lists
15. Find All Pairs with a Given Sum in a List
16. Merge Two Sorted Lists Without Duplicates

1

```
def fact(n):
    if n == 0 | n == 1:
        return 1
    else:
        return n * fact(n - 1)

n = int(input("Enter the number to find its factorial:"))
print(fact(n))
```

2

```
def reverse_num(n):
    rev = 0
    while n > 0:
        rem  = n % 10
        rev = rev * 10 + rem
```

```
        n = n // 10
    return rev

n = int(input("Enter the number to find its reverse:"))
print(reverse_num(n))

# 3
def print_odd(a : list):
    i = 1
    while i < len(a):
        print(a[i])
        i += 2

a = [5, 9, 2, 8, 0, 7]
print("Odd indexed elements in the list ", a)
print_odd(a)

# 4
def cube_nums(n):
    for i in range(1, n+1):
        print(i**3)

n = int(input("Enter the number to find cubes upto:"))
cube_nums(n)

# 5

def series_sum(n):
    s = 0
    for i in range(1, n+1):
        s += 1 / i
    return s

n = int(input("Enter the number upto which series sum req:"))
print(series_sum(n))

# 6
def smallest(lst):
    small = lst[0]
    for num in lst:
        if num < small:
```

```
            small = num
    return small

a = [4, 8, 9, 3, 7, 1]
print("Smallest in the list ", a)
print(smallest(a))

# 7
def remove_dupes(lst):
    res = []
    for num in lst:
        if num not in res:
            res.append(num)
    return res

a = [8, 3, 4, 2, 3, 2, 9, 7, 9]
print("After removing duplicates from the list ",a)
print(remove_dupes(a))

# 8
def remove_even(lst):
    return [x for x in lst if x % 2]

a = [4, 7, 9, 2, 3, 6]
print("List after removing even numbers:", remove_even(a))

# 9
def second_largest(lst):
    lst = sorted(lst)
    return lst[-2] if len(lst) > 1 else None

a = [5, 1, 8, 3, 9, 2]
print("Second largest element in the list:", second_largest(a))

# 10
def all_prime(lst):
    for n in lst:
        if n < 2:
            return False
        for i in range(2, int(n**0.5) + 1):
            if n % i == 0:
```

```
            return False
    return True

a = [3, 5, 7, 11]
print("All numbers are prime:", all_prime(a))

# 11
def freq_count(lst):
    freq = {}
    for num in lst:
        freq[num] = freq.get(num, 0) + 1
    return freq

a = [1, 2, 2, 3, 3, 3, 4]
print("Frequency of elements:", freq_count(a))

# 12
def longest_seq(lst):
    s, mx = set(lst), 0
    for num in lst:
        if num - 1 not in s:
            y = num
            while y in s:
                y += 1
            mx = max(mx, y - num)
    return mx

a = [100, 4, 200, 1, 3, 2]
print("Longest consecutive sequence length:", longest_seq(a))

# 13
def rotate(lst, k):
    k %= len(lst)
    return lst[-k:] + lst[:-k]

a = [1, 2, 3, 4, 5]
k = 2
print("List after rotating by", k, "positions:", rotate(a, k))

# 14
def intersection(lst1, lst2):
```

```
    res = []
    for num in lst1:
        if num in lst2 and num not in res:
            res.append(num)
    return res

a = [1, 2, 3, 4]
b = [3, 4, 5, 6]
print("Intersection of two lists:", intersection(a, b))

# 15
def pairs_with_sum(lst, target):
    seen, res = set(), []
    for num in lst:
        if target - num in seen:
            res.append((target - num, num))
        seen.add(num)
    return res

a = [1, 2, 3, 4, 5]
target = 6
print("Pairs with sum", target, ":", pairs_with_sum(a, target))

# 16
def merge_sorted(lst1, lst2):
    res = []
    i, j = 0, 0
    while i < len(lst1) and j < len(lst2):
        if lst1[i] < lst2[j]:
            res.append(lst1[i])
            i += 1
        elif lst1[i] > lst2[j]:
            res.append(lst2[j])
            j += 1
        else:
            res.append(lst1[i])
            i += 1
            j += 1
    while i < len(lst1):
        res.append(lst1[i])
        i += 1
```

```
    while j < len(lst2):
        res.append(lst2[j])
        j += 1
    return res

a = [1, 3, 5, 7]
b = [2, 3, 6, 8]
print("Merged sorted list without duplicates:", merge_sorted(a, b))
```

CODES:

```python
# 1
def fact(n):
    if n == 0 | n == 1:
        return 1
    else:
        return n * fact(n - 1)

n = int(input("Enter the number to find its factorial:"))
print(fact(n))

# 2
def reverse_num(n):
    rev = 0
    while n > 0:
        rem  = n % 10
        rev = rev * 10 + rem
        n = n // 10
    return rev

n = int(input("Enter the number to find its reverse:"))
print(reverse_num(n))

# 3
def print_odd(a : list):
    i = 1
    while i < len(a):
        print(a[i])
        i += 2

a = [5, 9, 2, 8, 0, 7]
print("Odd indexed elements in the list ", a)
print_odd(a)

# 4
```

```python
# 4
def cube_nums(n):
    for i in range(1, n+1):
        print(i**3)

n = int(input("Enter the number to find cubes upto:"))
cube_nums(n)

# 5

def series_sum(n):
    s = 0
    for i in range(1, n+1):
        s += 1 / i
    return s

n = int(input("Enter the number upto which series sum req:"))
print(series_sum(n))

# 6
def smallest(lst):
    small = lst[0]
    for num in lst:
        if num < small:
            small = num
    return small

a = [4, 8, 9, 3, 7, 1]
print("Smallest in the list ", a)
print(smallest(a))
```

```python
# 7
def remove_dupes(lst):
    res = []
    for num in lst:
        if num not in res:
            res.append(num)
    return res

a = [8, 3, 4, 2, 3, 2, 9, 7, 9]
print("After removing duplicates from the list ",a)
print(remove_dupes(a))

# 8
def remove_even(lst):
    return [x for x in lst if x % 2]

a = [4, 7, 9, 2, 3, 6]
print("List after removing even numbers:", remove_even(a))

# 9
def second_largest(lst):
    lst = sorted(lst)
    return lst[-2] if len(lst) > 1 else None

a = [5, 1, 8, 3, 9, 2]
print("Second largest element in the list:", second_largest(a))

# 10
def all_prime(lst):
    for n in lst:
        if n < 2:
            return False
        for i in range(2, int(n**0.5) + 1):
            if n % i == 0:
                return False
    return True
```

```python
a = [3, 5, 7, 11]
print("All numbers are prime:", all_prime(a))

# 11
def freq_count(lst):
    freq = {}
    for num in lst:
        freq[num] = freq.get(num, 0) + 1
    return freq


a = [1, 2, 2, 3, 3, 3, 4]
print("Frequency of elements:", freq_count(a))

# 12
def longest_seq(lst):
    s, mx = set(lst), 0
    for num in lst:
        if num - 1 not in s:
            y = num
            while y in s:
                y += 1
            mx = max(mx, y - num)
    return mx

a = [100, 4, 200, 1, 3, 2]
print("Longest consecutive sequence length:", longest_seq(a))

# 13
def rotate(lst, k):
    k %= len(lst)
    return lst[-k:] + lst[:-k]

a = [1, 2, 3, 4, 5]
k = 2
print("List after rotating by", k, "positions:", rotate(a, k))
```

```python
# 14
def intersection(lst1, lst2):
    res = []
    for num in lst1:
        if num in lst2 and num not in res:
            res.append(num)
    return res

a = [1, 2, 3, 4]
b = [3, 4, 5, 6]
print("Intersection of two lists:", intersection(a, b))

# 15
def pairs_with_sum(lst, target):
    seen, res = set(), []
    for num in lst:
        if target - num in seen:
            res.append((target - num, num))
        seen.add(num)
    return res

a = [1, 2, 3, 4, 5]
target = 6
print("Pairs with sum", target, ":", pairs_with_sum(a, target))
```

```
print("Pairs with sum", target, ":", pairs_with_sum(a, target))

# 16
def merge_sorted(lst1, lst2):
    res = []
    i, j = 0, 0
    while i < len(lst1) and j < len(lst2):
        if lst1[i] < lst2[j]:
            res.append(lst1[i])
            i += 1
        elif lst1[i] > lst2[j]:
            res.append(lst2[j])
            j += 1
        else:
            res.append(lst1[i])
            i += 1
            j += 1
    while i < len(lst1):
        res.append(lst1[i])
        i += 1
    while j < len(lst2):
        res.append(lst2[j])
        j += 1
    return res

a = [1, 3, 5, 7]
b = [2, 3, 6, 8]
print("Merged sorted list without duplicates:", merge_sorted(a, b))
```

OUTPUTS:

```
Enter the number to find its factorial:3
6
Enter the number to find its reverse:3
3
Odd indexed elements in the list  [5, 9, 2, 8, 0, 7]
9
8
7
Enter the number to find cubes upto:4
1
8
27
64
Enter the number upto which series sum req:3
1.8333333333333333
Smallest in the list  [4, 8, 9, 3, 7, 1]
1
After removing duplicates from the list  [8, 3, 4, 2, 3, 2, 9, 7, 9]
[8, 3, 4, 2, 9, 7]
List after removing even numbers: [7, 9, 3]
Second largest element in the list: 8
All numbers are prime: True
Frequency of elements: {1: 1, 2: 2, 3: 3, 4: 1}
Longest consecutive sequence length: 4
List after rotating by 2 positions: [4, 5, 1, 2, 3]
Intersection of two lists: [3, 4]
Pairs with sum 6 : [(2, 4), (1, 5)]
```

```
List after removing even numbers: [7, 9, 3]
Second largest element in the list: 8
All numbers are prime: True
Frequency of elements: {1: 1, 2: 2, 3: 3, 4: 1}
Longest consecutive sequence length: 4
List after rotating by 2 positions: [4, 5, 1, 2, 3]
Intersection of two lists: [3, 4]
Pairs with sum 6 : [(2, 4), (1, 5)]
Longest consecutive sequence length: 4
List after rotating by 2 positions: [4, 5, 1, 2, 3]
Intersection of two lists: [3, 4]
Pairs with sum 6 : [(2, 4), (1, 5)]
Merged sorted list without duplicates: [1, 2, 3, 5, 6, 7, 8]
PS C:\python>
```