

Assignment - 7

Page No.: 54

Q1: Write a function that removes duplicate values from a list.

//Program: def remove(lst):
 return list(set(lst))

//Output:
[1, 2, 3, 4, 5]

Q2: Write a function that returns cumulative sum of a list.

//Program: def cumlist(lst):

 s = 0
 res = []
 for i in lst:
 s += i
 res.append(s)

 return res

//Output: print(cumlist([1, 2, 3, 4]))

[1, 3, 6, 10]

Q3: Write a program to find frequency of each letter in a sentence.

//Program: sentence = "Vishat Kohli".

freq = {}
for ch in sentence:

 if ch != ":":

 freq[ch] = freq.get(ch, 0) + 1

//Output: print(freq)

{'V': 1, 'i': 2, 's': 1, 'h': 1, 'a': 1, 't': 1, 'K': 1, 'o': 1, 'l': 1,
' ': 1}

Q4: WAP to show the outputs!

Program:

```
(a) def fun1():
    l1 = list()
    l2 = list()
    for i in range(0, 5):
        l1.append(i)
        l2.append(i+3)
    print(l1)
    print(l2).
```

func1()

Output:

[0, 1, 2, 3, 4]
[3, 4, 5, 6, 7]

(b) def fun2():
 l1 = list()

```
    l2 = list()
    for i in range(0, 5):
        l1.append(i)
        l2.append(i+3)
```

l1, l2 = l2, l1

```
print(l1)
print(l2)
```

Output: func2()

[3, 4, 5, 6, 7]
[0, 1, 2, 3, 4]

Q5: WAP to show the outputs:

(a) c = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

```
result = 0
for i in range(0, 10):
    if (c[i] % 2 == 0):
        result += c[i]
```

print(result)

Output:

30

(b) $c = [1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$
 result = 0
 for i in range(0, 10):
 if (c[i] % 2 != 0):
 result += c[i]
 print(result)

app 25

(c) subject = 'computer'
 subject = list(subject)
 ch = subject[0]
 for i in range(0, len(subject) - 1):
 subject[i] = subject[i + 1]
 subject[len(subject) - 1] = ch
 print(''.join(subject))

app computer

(d) quantity = [15, 30, 12, 34, 56, 99]
 total = 0
 for i in range(0, len(quantity)): :
 if (quantity[i] > 15):
 total += quantity[i]
 print(total)

app 219

(e) x = [1, 2, 4, 6, 9, 10, 14, 15, 17]
 for i in range(0, len(x)): :
 if (x[i] % 2 == 0):
 x[i] = 4 * i
 elif (x[i] % 3 == 0):
 x[i] = 9 * i
 else:
 x[i] *= 2
 print(x)

app [2, 4, 8, 12, 36, 20, 24, 63, 34]

Q6: WAP to create a list of n lists containing first five multiples.

program: def multiples(n):
 return [i * j for i in range(1, n) for j in range(1, n+1)]

```
    print(multiples(3))
```

$$\left[[1, 2, 3, 4, 5], [2, 4, 6, 8, 10], [3, 6, 9, 12, 15] \right]$$

Q7: WAP to convert number into words using dictionary.

program:

```
d = { '0': 'zero', '1': 'one', '2': 'two', '3': 'three', '4': 'four',  
     '5': 'five', '6': 'six', '7': 'seven', '8': 'eight',  
     '9': 'nine' }
```

num = "452"

```
Jan[i] in num:  
print(q[i], end="")
```

app

four five two

Q8: WAP to find outputs:

Q8: WAP to find outputs:
programme: Address = '8-6, Lodhi road, Delhi'.
 $i \in \{1, 2, 3\}$

list1 = [1, 2, 3]

1872 = ['a', 1 , '2', 26 , 'd', 4]

tuple1 = ('a', 'e', 'i', 'o', 'u')

```
tuple2 = ([2,4,6,8], [3,6,9], [4,8], 5)
```

dict1 = { 'apple' : 'red', 'mango' : 'yellow', 'orange' : 'orange' }

dict2 = { 'x': ['eng', 'hind', 'maths', 'science'],

'xii': ['english', 'physics', 'chemistry',
'maths'] }

- (a) `list1[3]=4` // IndexError
 - (b) `print(list1 * 2)` // [1, 2, 3, 1, 2, 3]
 - (c) `print(min(list2))` // TypeError.
 - (d) `print(max(list1))` // 3
 - (e) `print(list(address))`
// ['B', 'C', 'G', 'I', 'L', 'O', 'D', 'H', '!', ' ', 'n', 'o', '
'a', 'q', ' ', ' ', ' ', 'D', 'e', 't', 'h', '!', 'J']

(f) list2. extend ([e', 5])

print(list2)

// [a', l, 'z', 26, 'd', 4, 'e', 5]

(g) list2. append ([e', 5])

print(list2)

// a', l, 26, 'd', 4, 'e', 5, [e', 5]

(h) names = ['mahan', 'mahan', 'gita']

names. sort (key=len)

print(names)

// ['gita', 'mahan', 'mahan']

(i) list3 = [(x*x) for x in range(1, 11)]

print(list3)

// [2, 4, 6, 8, 10, 12, 14, 16, 18, 20]

(j) del list3[1:]

print(list3)

// [2]

(k) list4 = [x+y for x in range(1, 5) for y in range(1, 5)]

print(list4)

// [2, 3, 4, 5, 3, 4, 5, 6, 4, 5, 6, 7, 5, 6, 7, 8]

(l) tuple2[3] = 6

// TypeError.

(m) tuple2.append(5)

// AttributeError:

(n) t1 = tuple2 + (5)

// TypeError.

(o) print('_'.join(tuple2));

// a-e-i-o-u

(p) print(list(zip(['apple', 'orange'], ('red', 'orange'))))

// [('apple', 'red'), ('orange', 'orange')]

(q) print(dict2['x11'])

// ['english', 'physics', 'chemistry', 'maths']

- (n) `dict2['XII'].append('computer science')
print(dict2)`
 // {('X'): ['eng', 'hind', 'maths', 'science'],
 // ('XII'): ['english', 'physics', 'chemistry', 'maths',
 // 'computer science']} }
- (o) `print('red' in dict1)`
 // False
- (p) `print(list(dict1.items()))`
 // [('apple', 'red'), ('mango', 'yellow'),
 // ('orange', 'orange')]
- (q) `print(list(dict2.keys()))`
 // ['X', 'XII']
- (r) `print(dict2.get('X1', 'None'))`
 // None.
- (s) `dict1.update({'kiwi': 'green'})
print(dict1)`
 // {'apple': 'red', 'mango': 'yellow', 'orange': 'orange',
 // 'kiwi': 'green'}

Q9: Determine the output of the following statements.
 program: vehicles = {'Bicycle', 'Scooter', 'Car', 'Bike', 'Truck',
 'Bus', 'Rickshaw'}

heavyVehicles = {'Truck', 'Bus'}

lightVehicles = {'Rickshaw', 'Scooter', 'Bike',

'Bicycle'}

(a) `lytVehicles = vehicles - heavyVehicles
print(lytVehicles)`

// { "Bicycle", "Scooter", "Car", "Bike", "Rickshaw"}

(b) `hvyVehicles = vehicles - lightVehicles
print(hvyVehicles)`

// { 'Truck', 'Bus', 'Car'}

(e) averageWeightVehicle = lightVehicles & heavyVehicles
 print(averageWeightVehicles)

// { 'Car' }

(f) transport = lightVehicles | heavyVehicles
 print(transport)

// { 'Rickshaw', 'Scooter', 'Bike', 'Bicycle', 'Truck',
 'Bus' }

(g) transport.add('Car')
 print(transport)

// { 'Rickshaw', 'Scooter', 'Bike', 'Bicycle', 'Truck',
 'Bus', 'Car' }

(h) for i in vehicles:
 print(i)

// Bicycle
 Scooter
 Car
 Bike
 Truck
 Bus
 Rickshaw

(i) print(len(vehicles))

// 7

(j) print(min(vehicles))

// Bicycle

(k) print(set.union(vehicles, lightVehicles, heavyVehicles))

// { 'Bicycle', 'Scooter', 'Car', 'Bike', 'Truck', 'Bus',
 'Rickshaw' }

Q10: WAP to print Recursive function to multiply two numbers.

//Program: def multiply(a, b):
 if $b = 0$:
 return 0
 return a + multiply(a, b-1)
 print(multiply(4, 5))

//app:

20

Q11: Recursive function to find Fibonacci number.

Program: def fib(n):
 if $n \leq 1$:
 return n
 return fib(n-1) + fib(n-2)

//app:

8

Q12: WAP to show Recursive binary search.

Program: def binSearch(arr, l, r, x):
 if $l \leq r$:
 mid = (l+r)//2
 if arr[mid] == x:
 return mid
 elif arr[mid] > x:
 return binSearch(arr, l, mid-1, x)
 else:
 return binSearch(arr, mid+1, r, x)

return -1

arr = [2, 4, 6, 8, 10]

print(binSearch(arr, 0, len(arr)-1, 8))

//app:

3

Q13: WAP to show recursive function to find length of a string.

Program: def length(s):

if s == "":

return 0

return 1 + length(s[1:])

print(length("Vishat"))

1/2p

5

Q14: WAP to recursive function to reverse a string.

Program: def rev(s):

if s == "":

return s

return rev(s[1:]) + s[0]

print(rev("Kashif"))

1/2p:

ikhok