Find outputs:

```
a = range(10,20)
print(*a, sep=',')  — # (10,11,12,13,14,15,16,17,18,19,20)

b = range(5)
print(*b) — #(0 1 2 3 4)

c = rang(10,1,-1)
print(*c, sep='...') — # 10...9...8...7...6...5...4...3...2

d = range(-10,0)
print(*d) — # -10,-9,-8,-7,-6,-5,-4,-3,-2,-1

e = range(-10)
print(*e) — # Empty listing

f = range(2,2)
print(*f) — # Empty string

r = range(10,17,3) — # [10,13,16]
a,b,c = r — # (10,13,16)

r = range(3) — # [0,1,2]
x,y,z = r — # Error
```

18/7/25

```
a = [25, 10.8, 'Hyd', True, 3+4j, None, 'Hyd', 25]
print(a) — [25, 10.8, 'Hyd', True, 3+4j, None, 'Hyd', 25]
print(*a) — 25 10.8 Hyd True 3+4j None Hyd 25
print(type(a)) — # class 'list'
print(id(a)) — # Address of object 'a'.
print(len(a)) — # 8
```

```
a[2] = 'Sec'
print(a) —— [25, 10.8, Sec, True, (3+4j), None, 'Hyd', 25]
print(a[2:5]) —— ['Sec', True, 3+4j']
```

Append & Remove

```
a = [ ]  —# [ ]
print(a)  #
a.append(25)
a.append(10.8)
a.append('Hyd')
a.append(True)
print(a)  —# [25, 10.8, Hyd, True]
a.remove('Hyd') —— # [25, 10.8, True]
  print(a)
a.remove('25')
  print(a)  —— # [ ~~10.8, True~~ ] Error
```

Find outputs:

```
a = [25, 10.8, 'Hyd']
print a  —# [25, 10.8, 'Hyd']
print id(a))  —— # Address of object 'a'
print(a*3)  —# [25,10.8,'Hyd',25,10.8,'Hyd',25,10.8,'Hyd']
print(a*2)  —# [25,10.8,'Hyd',25,10.8,'Hyd']
print(a*1)  —# [25, 10.8, 'Hyd']
print(a*0)  —# [ ] empty list
print(a*-1)  —— # [] empty list
print(a*4.0)  —— # Error.
```

```
a = a*3 ——# Ref modified to a[25, 10.5, Hyd, 25, 10.5, Hyd, 25, 10.5, Hyd]
print(a) —— [25, 10.5, 'Hyd', 25, 10.5, Hyd, 25, 10.5, Hyd]
print(id(a)) — Address of object 'a'
```

List ():

```
a = list('Hyd')
print(a) ——> ['Hyd'] ['H', 'y', 'd']
print(type(a)) —— # class <'list'>
print(len(a)) —— # 3 H, y, d
b = (10, 20, 15, 18)
print(list(b)) —— [10, 20, 15, 18]
print(list(range(5))) —— # [0, 1, 2, 3, 4]
print(list(25)) —— # Error.
```

Find outputs:

```
a = []
print(type(a)) —— # class <'list'>
print(a) —— # []
print(len(a)) —— # 0
b = list()
print(b) —— # []
print(len(b)) — 0
```

Find outputs:

```
a = [] — # Empty list
b = () — # Empty tuple
c = {} — # Empty set dict
d = set() — Empty set
print(type(a)) - <class 'list'>
print(type(b)) - <class 'tuple'>
print(type(c)) - <class 'dict'>
print(type(d)) - <class 'set'>
print(a) - []
print(b) - ()
print(c) - {}
print(d) - set()
```

Slice demo pgm:-

list = [25, 10.8, 3+4j, 'Hyd', True, None, 10.8, 'Hyd']
indexes: 0, 1, 2, 3, 4, 5, 6, 7
negative indexes: -8, -7, -6, -5, -4, -3, -2, -1

print(list[2:7]) — # [3+4j, Hyd, True, None, 10.8]

print(list[::]) — # [25, 10.8, 3+4j, Hyd, True, None, 10.8, Hyd]

print(list(::-1)) [-1:-9:-1] => -1 to -8 — # Reverse of list

print(list[::2]) (0:8:2) — # [25, 3+4j, True, 10.8]

print(list(1::2) (1:9:2) — # [10.8, Hyd, None, Hyd]

print(list(::-2] — # [Hyd, None, Hyd, 10.8]

print(list[-2::-2] [-2:-9:-2] — # [10.8, True, 3+4j, 25]

print(list[1:4]) — # [10.8, 3+4j, Hyd]

print(list[-4:-1]) — # [True, Hyd, None, 3+4j, 10.8, 25]

print(list[3:-3]) 3:-4:1 — # [Hyd, True, None]

print(list[2:-5]) 2:-6:1 — # [] [3+4j]

print(list[-1:-5] — # []

Find output:

list = [25, 10.8, 3+4j, 'Hyd', True, None, 10.8, 'Hyd']

a, y = list[3:5] — # ['Hyd', True]

print('x:', x) — # Hyd

print('y:', y) — # True

Print set in 2 diff. ways:

```
a = {25, True, 'Hyd', 10.8}
print ('set with print function:') —# {25, True, 'Hyd', 10.8}
print (a)

print ('Iterate through set with for loop:')
    for element in a:        —#  25
        print(element)           Hyd
                                 10.8
                                 True
```

Tricky program:

```
a = set ()      — # Empty Set
a.add (25)      — # (25,)
a.add (10.8)    — # (25, 10.8)
a.add ('Hyd'    — # (25 10.8 'Hyd')
a.add (True)    — # (25 'Hyd' 10.8 1)
a.add (None)    — # (None 10.8 25 'Hyd')
a.add (1)       — # (1, 10.8, 25, 'Hyd', 25)
print (a)       — # (1, 25, 10.8, ~~True~~ 1 'Hyd' None)
a.remove (25)   — # (1, 10.8, 1, Hyd', None)
a.append (100)  — # No append method so Error.
a.add(set ())   — # Error set invalid set (Nested set)
                    not valid.
a.add ([])      — # Error list cannot be valid as
                    it is mutable.
```