```
# 1
l1 = [1,3,5,7,9]
print(l1)
l1[(len(l1)-1)],l1[0] = l1[0],l1[(len(l1)-1)]
print(l1)


# 2
lst = [1,2,3,4,5,6,7,8,]
lst1=[]
lst2=[]
for i in range(0,len(lst)):
  if i%2 ==0:
    lst1.append(lst[i])
  else:
    lst2.append(lst[i])

print(lst)
print(lst1)
print(lst2)


# 3
l1 = [2,5,2,3,4,6,3,9,5,4]
l2 = [3,4]

result = False
for i in range(len(l1)):
    if l1[i:i+len(l2)] == l2:
        result = True
        break  # Exit loop once subsequence is found

print(result)



# 4
l1 = [1,3,5,7,9,11,13]
l2 = [2,4,6]
l3 =[]
for i in range(len(l1)):
  l3.append(l1[i])
  for j in range(i,len(l2)):
    l3.append(l2[j])
    break

print(l3)


# 5

# N/A


# 6
tup = ('x','y','z')
lst = list(tup)
index_pos = int(input("Index: "))
element = (input("Enter the element: " ))
lst.insert(index_pos,element)
new_tup = tuple(lst)
print(new_tup)
```

```
# 7
tup = (1,1,1,1,1,1,1,1)
nth_element = 3
lst = list(tup)
for i in range(nth_element - 1, len(lst), nth_element):
  lst[i] = lst[i]*3

print(lst)


# 8
tup = ('d','c','b','a')
tup1 =()
for i in range(len(tup)-1,-1,-1):
  tup1 += (tup[i],)
print(tup1)


# 9

# N/A


# 10
tups =[(1, 2, 3), (4, 5, 6), (7, 8, 9), (10, 11, 12)]
for i in tups:
  print((i[0],i[-1]),end =" ")

#x = tuple((tup[0],tup[-1]) for tup in tups )
#print(x)


# 11

dct = {'Iphone':'Puppy','IQOO': 'Venkat', 'OnePlus': 'VK'}
sub_val = ['Venkat','VK']
extracted_keys = []
for keys,values in dct.items():
  if values in sub_val:
    extracted_keys.append(keys)
print(extracted_keys)


# 12
dict1= {1:'a',2:'b',3:'c',4:'d'}
x = dict1.keys()
y = dict1.values()
vals = []
keys =[]
for i in x:
  vals.append(i)
for i in y:
  keys.append(i)

new_dict = dict(zip(keys,vals))
print(new_dict)


# 13

# N/A
```

```
# 14
def partition_values_by_modulus(dictionary, modulus):
    return {i: [value for value in dictionary.values() if value % modulus == i] for i in range(modulus)}

# Example usage:
my_dict = {'a': 10, 'b': 7, 'c': 15, 'd': 23, 'e': 8}
modulus = 3
partitioned_dict = partition_values_by_modulus(my_dict, modulus)
print("Partitioned dictionary:", partitioned_dict)


# 15

# N/A


# 16
set1 = {1,2,3,4,5}
set2 = {4,5,6,7,8}
x = (set1 | set2) - (set1 & set2)
print(x)


# 17
set1 = {1,2,3,4,5,6,7}
set2 = {2,6}
set3=set()
for i in set2:
  for j in set1:
    if i == j:
      set3.add(i)

if set2 == set3:
  print(True)
else:
  print(False)
#x = set2<=set1
#print(x)


# 18
set_a = {1,2,3}
set_b = {'x','y','z'}
pairs =[]
for i in set_a:
  for j in set_b:
    if i !=j:
      pairs.append((i,j))
for i in pairs:
  print(i)


# 19
set_a = {2,4,6,8}
set_b = {1,2,3,1}
x = [i for i in set_a if i not in set_b]
print(set(x))
```

```
# 20
def find_subsets(input_set, divisor):
    subsets = []
    for num in input_set:
        if num % divisor == 0:
            subsets.append({num})
    return subsets

# Example usage:
input_set = {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
divisor = 2
result = find_subsets(input_set, divisor)
print("Minimum set of subsets where each subset is divisible by", divisor, ":")
for subset in result:
    print(subset)



# 21
from itertools import zip_longest

def interleave(*lists):
    interleaved = []
    for elements in zip_longest(*lists):
        interleaved.extend([elem for elem in elements if elem is not None])
    return interleaved

list1 = [1, 2, 3]
list2 = ['a', 'b', 'c', 'd']
list3 = [True, False]
result = interleave(list1, list2, list3)
print("Interleaved list:", result)


# 22

lst_strs = ['LV','DSAI','HIVE']
str_len =[]
for i in lst_strs:
  str_len.append(len(i))

dct = dict(zip(lst_strs,str_len))
print(dct)

''' --- using fuction ---
def str_dict(keys,values):
  str_len =[]
  for i in lst_strs:
    str_len.append(len(i))
    dct = dict(zip(lst_strs,str_len))
  print(dct)
lst_strs = ['LV','DSAI','HIVE']
str_dict(lst_strs,str_len)
'''
```

```
# 23
def fib(n):
  a=0
  b=1
  if n <= 0:
    return 0
  elif n == 1:
    return a
  else:
    print(a)
    print(b)
    for i in range(2,n):
      c = a+b
      a=b
      b=c
      print(c)

fib(4)


# 24

l1 = [1,2,3,4,5,6]
n = 9
l2=[]
for i in range(len(l1)):
  for j in range(i,len(l1)):
    if l1[i]+l1[j] == n:
      l2.append((l1[i],l1[j]))

print(l2)



# 25

l1 = ['a','b','c','d','e']
k = 11
k = k % len(l1)
print(l1[k:]+l1[:k])




# 26

tups =((1,2,3),(4,5),(6,(7,9)))
sum = 0
for item in tups:
  for element in item:
    try:
      for sub_element in element:
        sum += sub_element
    except:
      sum += element
print(sum)


# 27

lst = [1,2,3,4,5,6,7,8,9]
n =3
tup = []
for i in range(0,len(lst),n):
  tup.append(tuple(lst[i:i+n]))
print(tuple(tup))
```

```
# 28

tups = ((1,2,3),(4,5,66),(7,8,9),(0,7,3))
max_sum = 0
max_tup = None
for item in tups:
  current_sum = 0
  for element in item:
    current_sum += element
  if current_sum > max_sum:
    max_sum = current_sum
    max_tup = item
print(max_tup)


# 29
tup1 =(1,2,3,4)
tup2=(5,6)
for i in tup1:
  if i in tup2:
    print(True)
    break
  print(False)
  break

'''
def common_elements(tup1, tup2):
    for element in tup2:
        if element in tup1:
            return True
    return False
tup1 = (1, 2, 3, 4)
tup2 = ( 5, 6)

result = common_elements(tup1, tup2)
print(result)
'''


# 30

tup1 = (1,2,3,4,5)
sum = 0
new_tup = []
for i in tup1:

  sum += i
  new_tup.append(sum)
print(tuple(new_tup))

     (1, 3, 6, 10, 15)


# 31

# N/A
```

```
# 32

# 33

dict1 = {'a':10,'b':20,'c':40,'d':30}
max_val = max(dict1.values())
for key,value in dict1.items():
  if value == max_val:
    print(key)




# 34

# N/A


# 35

dict1 = {1:"DSAI", 2: "LV",3:"THE HIVE"}
sd = dict(sorted(dict1.items(),key = lambda item : len(item[1])))
print(sd)


# 36

set1 ={1,2,3}
set2 = {3,4,5}
set3 = {5,6,7}
union_set = set1 | set2 | set3
print(union_set)


# 37

set1 = {1,2,2,3}
set2 = {3,5,6}
print(set1.isdisjoint(set2))


# 38

lst_sets= [ {1,2,3},{4,5,6},{7,8,9} ]
result = set()
for i in lst_sets:
  result = result.symmetric_difference(i)
print(result)
```