

ASSIGNMENT-5

# PYTHON

# **NAME : MADA SRAGVIN KUMAR**

# **MIS NO : 112315097**

# **GROUP : 3**

**YEAR : 2**

**SECTION : A**

# **7 of assignment 4**

n=int(input("enter the no of items u want to add : "))

l=[]

for i in range(0,n):

c=[]

k=input(f"enter the details of {i+1} person: ")

c=k.split(",")

c[2]=int(c[2])

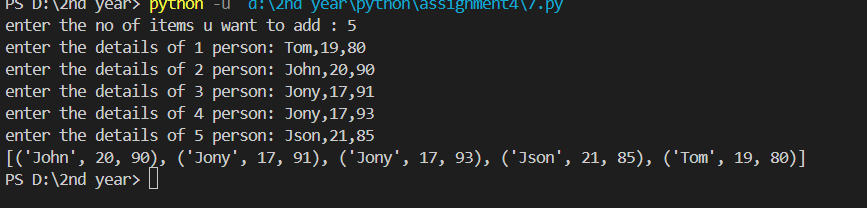
c[1]=int(c[1])

l.append(tuple(c))

import operator

l.sort(key=operator.itemgetter(0,1,2))

print(l)



# **9 of assignment 4**

a=input("enter the string: ")

b=a.split()

k=[]

d={}

b.sort()

for i in range(0,len(b)):

if i not in b:

if b[i].isnumeric():

d.update({b[i]:b.count(i)})

k.append(i)

for i in range(0,len(b)):

if i not in b:

if b[i][0].isnumeric():

d.update({b[i]:b.count(i)})

k.append(i)

for i in range(0,len(b)):

if i not in b:

d.update({b[i]:b.count(b[i])})

k.append(i)

print(d)

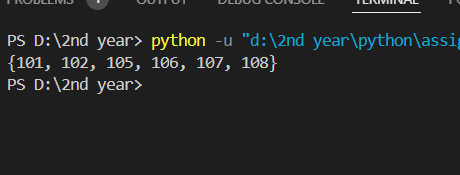


# **1**

store1\_sales=set([101,102,103,104,105])

store2\_sales=set([103,104,106,107,108])

print(store1\_sales.symmetric\_difference(store2\_sales))



# **2**

def intersect (a,b,c):

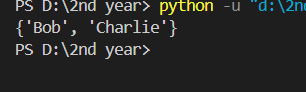
return a&b&c

facebook\_friends={"Alice","Bob","Charlie"}

twitter\_friends={"Bob","Charlie","David"}

linkedin\_friends={"Bob","Charlie","Emma"}

print(intersect(facebook\_friends,twitter\_friends,linkedin\_friends))



# **3**

n=int(input("enter the length of the string: "))

l=[]

for i in range(0,n):

k=int(input("enter the elements: "))

l.append(k)

b=[]

l.sort()

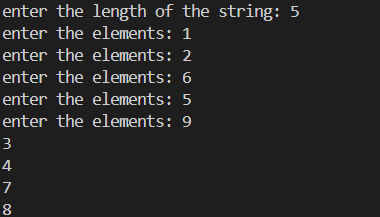
min=l[0]

max=l[n-1]

for i in range(min,max+1):

if i not in l:

print(i)



# **4**

n=int(input("enter the length of the string: "))

l=[]

for i in range(0,n):

k=(input("enter the elements: "))

l.append(k)

dup=set()

oneset=set()

for j in l:

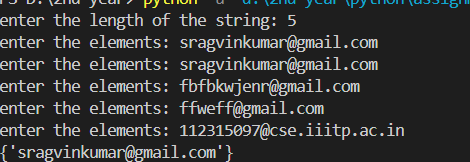
if j not in oneset:

oneset.add(j)

else:

dup.add(j)

print(dup)

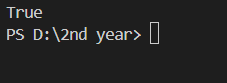


# **5**

a={0,1,2,3,4,5,6,7,8,9}

b={1,6,7,2,4}

print(b.issubset(a))



# **6**

set1={1,5,7,3,9,4,2}

set2={1,3}

set3={2,3}

set4={6,4}

l=[set1,set2,set3,set4]

k=int(input("enter the number: "))

setsub=set()

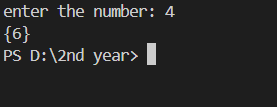
for i in l:

if i != l[k-1]:

setsub.update(i)

set=l[k-1]-setsub

print(set)



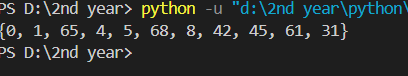
# **7**

a={1,5,1,0,8,45,61}

b={4,68,31,8,42,65}

a|=b

print(a)



# **8**

def dfs(node,G):

setg=set()

if G[node]==set():

return set()

for i in G[node]:

setg.update(i)

for j in G[node]:

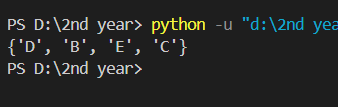
setg.update(dfs(j,G))

return setg

graph={'A':{'B','C'},'B':{'D'},'C':{'D'},'D':{'E'},'E':set()}

setf=dfs('A',graph)

print(setf)



# **9**

def has\_cycle(graph):

def dfs(node, visited, rec\_stack):

visited.add(node)

rec\_stack.add(node)

for neighbor in graph.get(node, []):

if neighbor not in visited and dfs(neighbor, visited, rec\_stack):

return True

elif neighbor in rec\_stack:

return True

rec\_stack.remove(node)

return False

visited = set()

for node in graph:

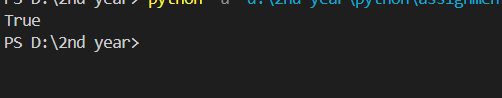
if node not in visited and dfs(node, visited, set()):

return True

return False

graph={"A":{"B"},"B":{"C"},"C":{"A"},"D":{"E"},"E":set()}

print(has\_cycle(graph))



# **10**

a=input("Enter the element of set with space betweem them: ")

s=set(a.split(" "))

def powerset(s):

s = list(s)

result = [[]]

for elem in s:

result += [subset + [elem] for subset in result]

return result

print(f"The powerset of {s} are: ")

print(powerset(s))

