**Question 1’s ,Answer:**

def checkpass(password):

upper\_alp="QWERTYUIOPASDFGHJKLZXCVBNM"

lower\_alp="qwertyuiopasdfghjklzxcvbnm"

number\_1="1234567890"

special\_char="!@#$%^&\*()"

if len(password)<8:

return False

if not any(char in upper\_alp for char in password):

return False

if not any(char in lower\_alp for char in password):

return False

if not any(char in number\_1 for char in password):

return False

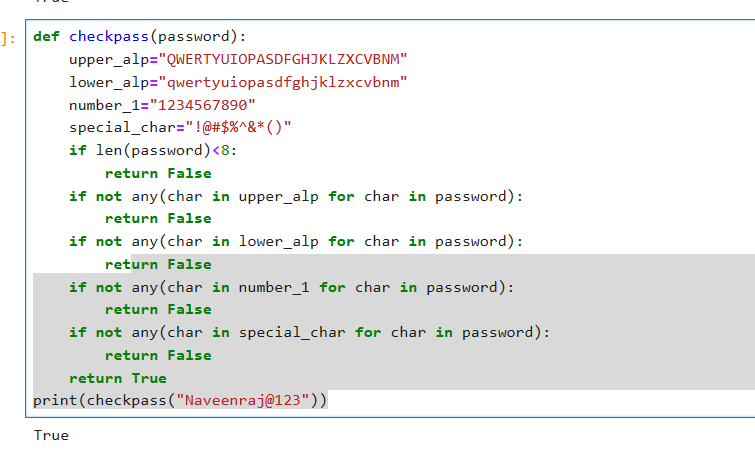
if not any(char in special\_char for char in password):

return False

return True

print(checkpass("Naveenraj@123"))

**output screen shot:**



**Question 2’s answer:**

def attendance(record):

record\_1=record.count('P')

record\_2=(record\_1/10)

if record\_2>=0.7:

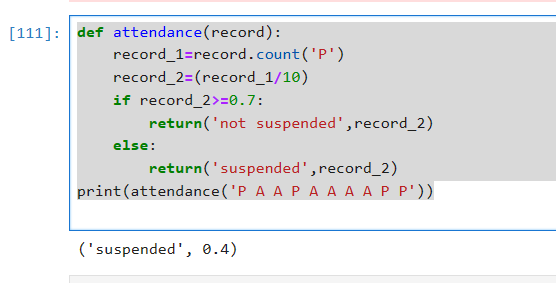
return('not suspended',record\_2)

else:

return('suspended',record\_2)

print(attendance('P A A P A A A A P P'))

**output evidence:**



**Question 3’s answers:**

def ankit\_max(marks\_dict):

if len(marks\_dict)<2:

return None

sort\_marks=sorted(marks\_dict.values(),reverse=True)

first\_marks=sort\_marks[0]

second\_marks=sort\_marks[1]

if first\_marks==second\_marks:

return None

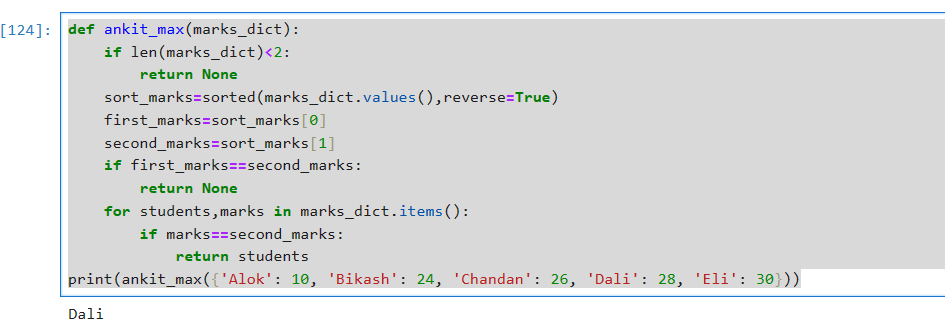
for students,marks in marks\_dict.items():

if marks==second\_marks:

return students

print(ankit\_max({'Alok': 10, 'Bikash': 24, 'Chandan': 26, 'Dali': 28, 'Eli': 30}))

**output evidence:**



def ankit\_max(marks\_dict):

if len(marks\_dict)<2:

return None

sort\_marks=sorted(marks\_dict.values(),reverse=True)

first\_marks=sort\_marks[0]

second\_marks=sort\_marks[1]

if first\_marks==second\_marks:

return None

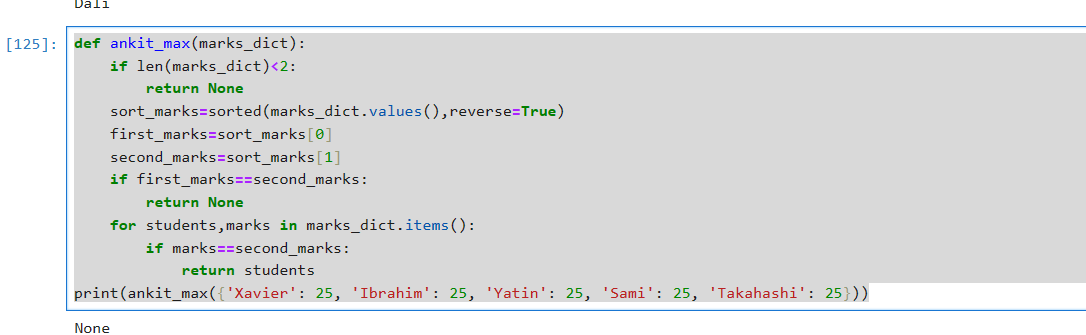
for students,marks in marks\_dict.items():

if marks==second\_marks:

return students

print(ankit\_max({'Xavier': 25, 'Ibrahim': 25, 'Yatin': 25, 'Sami': 25, 'Takahashi': 25}))

**output evidence:**



**Question 4’s answer:**

def multiple\_of\_3(number):

number\_1=str(number)

update\_num=[int(digi) for digi in number\_1]

sum\_num=sum(update\_num)

if sum\_num%3==0:

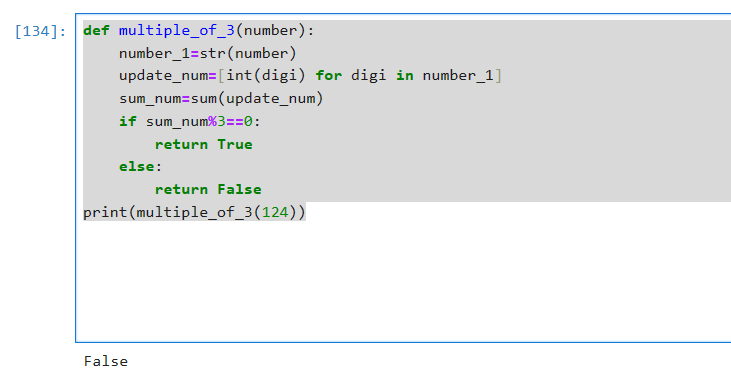
return True

else:

return False

print(multiple\_of\_3(124))

**output evidence:**



**Question 4’s answer:**

def multiple\_of\_3(number):

number\_1=str(number)

update\_num=[int(digi) for digi in number\_1]

sum\_num=sum(update\_num)

if sum\_num%3==0:

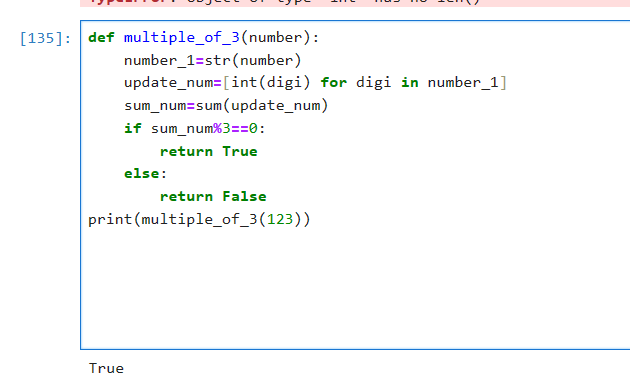
return True

else:

return False

print(multiple\_of\_3(123))

**output evidence:**



**Question 5’s answer**

def deduplicate(arg):

output\_value=''

previous\_value=None

for char in arg:

if char!=previous\_value:

output\_value+=char

previous\_value=char

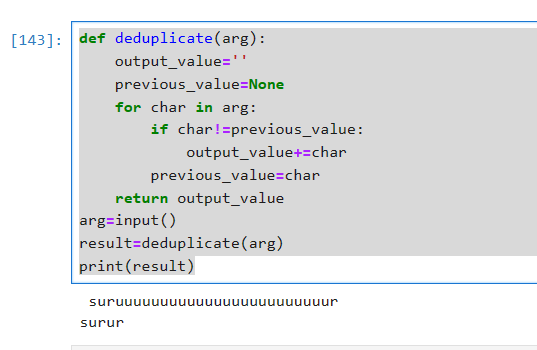
return output\_value

arg=input()

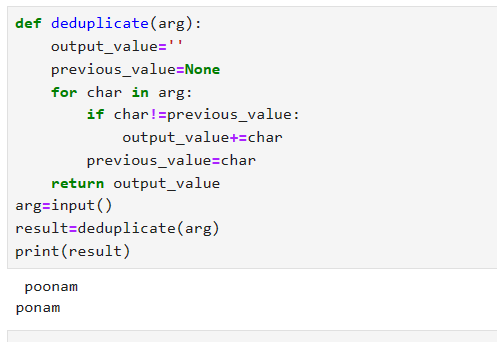
result=deduplicate(arg)

print(result)

**ouput evidence:**



**ouput evidence:**



**ouput evidence:**

