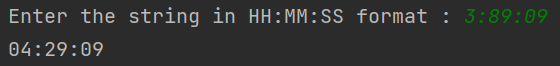
## 1. Correct the malformed time string for e.g. "5:70:65 to 6:11:05" Code:

# Correct the malformed time string for e.g. "5:70:65 to 6:11:05"  
time\_string = input("Enter the string in HH:MM:SS format : ")  
time\_list = time\_string.split(':')  
hours = int(time\_list[0])  
minutes = int(time\_list[1])  
seconds = int(time\_list[2])  
if seconds > 59:  
 minutes = minutes + seconds//60  
 seconds = seconds % 60  
else:  
 seconds = seconds  
if minutes > 59:  
 hours = hours + minutes//60  
 minutes = minutes % 60  
else:  
 minutes = minutes  
print(str(hours).zfill(2)+':'+str(minutes).zfill(2)+':'+str(seconds).zfill(2))

## Output:

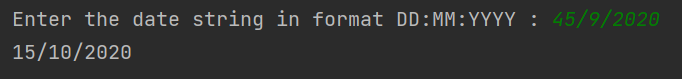


## 2. Correct the malformed date string for e.g. "45/8/2018" to "14/9/2018"

## Code:

# Correct the malformed date string for e.g. "45/8/2018" to "14/9/2018"  
date\_string = input("Enter the date string in format DD:MM:YYYY : ")  
date\_list = date\_string.split('/')  
dd = int(date\_list[0])  
mm = int(date\_list[1])  
year = int(date\_list[2])  
t1m = (mm == 1 or mm == 3 or mm == 5 or mm == 7 or mm == 8 or mm == 10 or mm == 12)  
t0m = (mm == 4 or mm == 6 or mm == 9 or mm == 11)  
Feb = (mm == 2)  
looper = True  
while looper:  
 if (year % 4 == 0 and year % 100 != 0) or (year % 400 == 0):  
 leap\_year = 1  
 else:  
 leap\_year = 0  
 special = (Feb and leap\_year and dd > 29) or (Feb and (not leap\_year) and dd > 28)  
 while mm <= 12 and ((t1m and dd > 31) or (t0m and dd > 30) or special):  
 if t1m:  
 dd -= 31  
 mm += 1  
 elif t0m:  
 dd -= 30  
 mm += 1  
 else:  
 if leap\_year:  
 dd -= 29  
 mm += 1  
 else:  
 dd -= 28  
 mm += 1  
 t1m = (mm == 1 or mm == 3 or mm == 5 or mm == 7 or mm == 8 or mm == 10 or mm == 12)  
 t0m = (mm == 4 or mm == 6 or mm == 9 or mm == 11)  
 Feb = (mm == 2)  
 special = (Feb and leap\_year and dd > 29) or (Feb and (not leap\_year) and dd > 28)  
 if mm > 12:  
 mm -= 12  
 year += 1  
 looper = True  
 else:  
 looper = False  
print(str(dd).zfill(2) + '/' + str(mm).zfill(2) + '/' + str(year).zfill(4))

## Output:

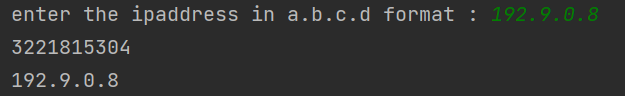


## 3. Convert ip address from "a.b.c.d" format to integer format and vice versa

## Code:

# Convert ip address from "a.b.c.d" format to integer format and vice versa  
ip\_string = input('enter the ipaddress in a.b.c.d format : ')  
my\_list = ip\_string.split('.')  
a = int((my\_list[0]))  
b = int((my\_list[1]))  
c = int((my\_list[2]))  
d = int((my\_list[3]))  
number = a\*256\*\*3 + b\*256\*\*2 + c\*256\*\*1 + d  
print(number)  
d = number % 256  
c = (number//256) % 256  
b = (number//(256\*\*2)) % 256  
a = (number//(256\*\*3)) % 256  
print(str(a)+"."+str(b)+"."+str(c)+"."+str(d))

## Output:

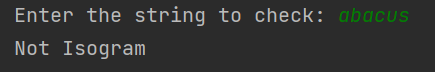


## 4. Check whether the given string is isogram or not.

## Code:

# Check whether the given string is isogram or not.  
ip\_string = input("Enter the string to check: ")  
for i in range(len(ip\_string)):  
 for a in range(i+1, len(ip\_string)):  
 if ip\_string[i] == ip\_string[a]:  
 print("Not Isogram")  
 quit()  
print("Isogram")

## Output:

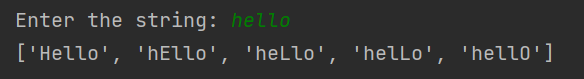


## 5. Given a string find a Mexican wave

## Code:

# Given a string, find a mexican wave  
ip\_string = input("Enter the string: ")  
listo = []  
for i in range(len(ip\_string)):  
 op\_string = ip\_string[:i] + ip\_string[i].upper() + ip\_string[i+1:]  
 listo.append(op\_string)  
print(listo)

## Output:

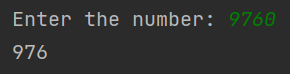


## 6. Given a number, find the largest number by deleting single digit (order of digits will remain same)

## Code:

# Given a number, find the largest number by deleting single digit(order of digits will remain same)  
number = input("Enter the number: ")  
number = int(number)  
greatest = 0  
listNum = []  
for i in range(1, len(str(number))+1):  
 a = ((number//(10\*\*i)) \* (10 \*\* (i-1))) + (number % (10 \*\* (i-1)))  
 listNum.append(a)  
for nums in listNum:  
 if nums > greatest:  
 greatest = nums  
print(greatest)  
# Given a number, find the largest number by deleting single digit(order of digits will remain same)  
number = input("Enter the number: ")  
number = int(number)  
greatest = 0  
listNum = []  
for i in range(1, len(str(number))+1):  
 a = ((number//(10\*\*i)) \* (10 \*\* (i-1))) + (number % (10 \*\* (i-1)))  
 listNum.append(a)  
for nums in listNum:  
 if nums > greatest:  
 greatest = nums  
print(greatest)

## Output:

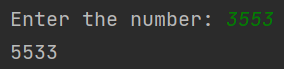


## 7. Given a number, find the largest number by shuffling digits

## Code:

number = input("Enter the number: ")  
number = int(number)  
List = []  
output = 0  
for i in range(len(str(number))):  
 a = ((number % (10 \*\* (i + 1))) - (number % (10 \*\* i))) // (10 \*\* i)  
 List.append(a)  
List.sort(reverse=True)  
for num in List:  
 output = (output \* 10) + num  
print(output)

## Output:

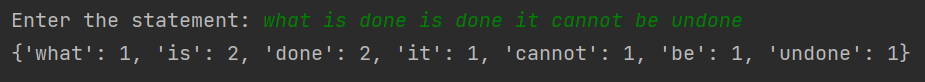


## 8. Compute the word frequency in the given message

## Code:

statement = input("Enter the statement: ")  
statement = statement.lower()  
wordList = statement.split(' ')  
dictionary = {}  
for word in wordList:  
 if word in dictionary.keys():  
 dictionary[word] += 1  
 else:  
 dictionary[word] = 1  
print(dictionary)

## Output:

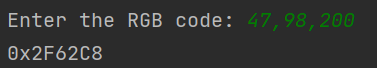


## 9. RGB to Hex conversion and vice versa for e.g. (255,0,255) into 0xFF00FF

## Code:

rgb = input("Enter the RGB code: ")  
codeList = rgb.split(',')  
conversion\_table = ['0', '1', '2', '3', '4', '5', '6', '7', '8', '9', 'A', 'B', 'C', 'D', 'E', 'F']  
final = ""  
for code in codeList:  
 decimal = int(code)  
 hexadecimal = ''  
 if decimal > 0:  
 while decimal > 0:  
 remainder = decimal % 16  
 hexadecimal = conversion\_table[remainder] + hexadecimal  
 decimal = decimal // 16  
 elif decimal == 0:  
 hexadecimal = '00'  
 final = final + hexadecimal  
print('0x' + final)

## Output:

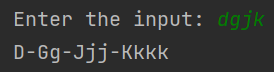


## 10. Generate accumulated strings, e.g. abcd ==> A-Bb-Ccc-Dddd

## Code:

string = input("Enter the input: ")  
charList = []  
for i in range(len(string)):  
 charList.append(string[i])  
output = []  
for index in range(len(charList)):  
 output.append(charList[index].upper())  
 loop = index  
 while loop > 0:  
 output.append(charList[index])  
 loop -= 1  
 if index < len(charList) - 1:  
 output.append("-")  
outputString = ""  
for i in range(len(output)):  
 outputString = outputString + output[i]  
print(outputString)

## Output:



## 1. In a given list of elements, all elements are equal except the one. Write a code to find the odd man out

## Code:

listEle = [24,24,24,25,24,24,24,24]  
if len(listEle) == 1:  
 print(None)  
listEle.sort()  
if listEle[0] != listEle[1]:  
 if listEle[0] == listEle[2]:  
 print(listEle[1])  
 else:  
 print(listEle[0])  
else:  
 print(listEle[-1])

## Output:



## 2. In a given list of elements, find the elements which is close to its mean

## Code:

listEle = [25,23,20,1,5,6]  
summ = 0  
for x in listEle:  
 summ += x  
mean = summ/len(listEle)  
diff = 9223372036854775807   
no = listEle[0]  
for x in listEle:  
 if abs(x - mean) < diff:  
 diff = abs(x-mean)  
 no = x  
print(no)

## Output:

## 

## 3. Find the average speed of vehicle, given the distance travelled for fixed time intervals, e.g. [0, 0.1, 0.25, 0.45, 0.55, 0.7, 0.9, 1.0]

## Code:

listi = [0, 0.1, 0.25, 0.45, 0.55, 0.7, 0.9, 1]  
sum = 0  
for x in listi:  
 sum += x  
print(round(sum/len(listi), 4))

## Output:

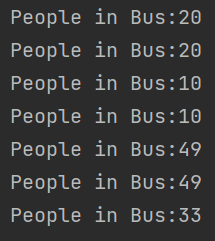


## 4. Find the no. of people in a bus, given data of people onboarding and alighting at each station

## Code:

onboard = [43, 23, 33, 14, 65, 6, 1]  
alight = [23, 23, 43, 14, 26, 6, 17]  
passengers = 0  
for x in range(len(onboard)):  
 passengers += onboard[x] - alight[x]  
 print(passengers)

## Output:



## 5. Find the missing number, given the original list and modified one

## Code:

origlist = [1,2,3,4,5,6,7,8]  
modded = [1,2,3,4,5,6,7]  
for x in origlist:  
 if x in modded:  
 continue  
 else:  
 print("Missing Number:"+str(x))

## Output:



## 6. Find the difference between two lowest numbers in the list

## Code:

newlist = [1,2,3,4,5,6]  
if len(newlist) < 2:  
 print(None)  
 quit()  
newlist.sort()  
print(newlist[1]-newlist[0])

## Output:



## 7. In a given a list, count no. of elements smaller than their mean

## Code:

listi = [1, 33, 3, 33, 53, 76]  
sum = 0  
for x in listi:  
 sum += x  
mean = sum/len(listi)  
listi.sort()  
count = 0  
for a in listi:  
 if a < mean:  
 count += 1  
 else:  
 break  
print(count)

## Output:

