###############################################################################################################################

###############################################################################################################################

Q1) '''The length of a month varies from 28 to 31 days. In this exercise you will create a program that reads the name of a month from the user as a string. Then your program should display the number of days in that month. Display “28 or 29 days” for February so that leap years are addressed'''

###############################################################################################################################

month\_dic t= {'JANUARY':31,'MARCH':31,'MAY':31,'JULY':31,'AUGUST':31,'OCTOBER':31, # Dictionary of month name as key and days as value

'DECEMBER':31,'APRIL':30,'JUNE':30,'SEPTEMBER':30,'NOVEMBER':30,'FEBRUARY':'28 or 29'}

month=input("Enter the any month name - ") #input in string for month variable

month\_upper=month.upper() #converting the input string in upper case form comparision and assign data to month\_upper

if(month\_upper in month\_dict): # matching month\_upper value in the dictionary of months keys

print(month\_dict[month\_upper],"days") #printing data

else:

print("please enter any month name") #if user is entering any other string, integer except month display this statement

###############################################################################################################################

Output:

Text

Description automatically generated

Explanation of logic: For this problem the most suitable data structure is Dictionary.

1. We are storing the month name(upper case) as key and days as values. So, we can search with month name and display a values (days)
2. Now, we are taking string input from user in month variable.
3. User can give input in any case hence we are converting user input in upper case for smooth comparison. Using(.upper())
4. Furthermore, we are applying **if**/**else** condition “ searching the input in dictionary “keys” if it matches it will display the value days else it will display message ("please enter any month name")

###############################################################################################################################

Q2) '''Write a program that reads a sound level in decibels from the user. If the user enters a decibel level that matches one of the noises in the table then your program should display a message containing only that noise. If the user enters a number of decibels between the noises listed then your program should display a message indicating which noises the level is between. Ensure that your program also generates reasonable output for a value smaller than the quietest noise in the table, and for a value larger than the loudest noise in the table'''

###############################################################################################################################

noise\_decibel={130: 'Jackhammer', 106: 'Gas lawnmower', 70: 'Alarm clock', 40: 'Quiet room'} #Dictionary of noise decibel as key and noise name as Values

noise=int(input('Enter the decibel - ')) #taking input from user in integer

if(noise in noise\_decibel): #matching noise value in the dictionary of noise\_decibel keys.

print(noise\_decibel[noise]) #printing the type of noise according to user's input

elif (130 < noise): # checking if noise decibel is greater than 130.

print('noise is greater than ', noise\_decibel[130])

elif (40 > noise): # checking if noise decibel is lesser than 40.

print('noise is lesser than ', noise\_decibel[40])

elif ((130 > noise) and (noise > 106)): # checking if noise decibel is between 130 and 106

print('noise is between', noise\_decibel[130], ' and ', noise\_decibel[106]) #printing the type of noise according to user's input

elif ((106 > noise) and (noise > 70)): # checking if noise decibel is between 106 and 70

print('noise is between', noise\_decibel[106], ' and ', noise\_decibel[70]) #printing the type of noise according to user's input

elif ((70 > noise) and (noise > 40)): # checking if noise decibel is between 70 and 40

print('noise is between', noise\_decibel[70], ' and ', noise\_decibel[40]) #printing the type of noise according to user's input

###############################################################################################################################

Output:

Text

Description automatically generated

Explanation of logic: For this problem the most suitable data structure is Dictionary.

1. We are storing the decibel as keys and noise name as values in dictionary. So, we can search with decibels and display a values (noise name)
2. Taking integer input from user in in noise variable for further use.
3. Now, search the input(noise) in dictionary key using **if/else** condition.
4. If its exact match, then it will print the exact noise name
5. However, if the input lies between two noise decibels e.g. ((70 > noise) and (noise > 40), then it will print both the nearest noise names, before and after that point (print 70 and 40 noise name).
6. If the input is greater than the highest decibels it will print “'noise is greater than the “noise name”(code will fetch this noise name from dict).
7. Similarly, If the input is lesser than the lowest decibels it will print “'noise is lesser than the “noise name”( code will fetch this noise name from dict).).