

```

1 #QUESTION-1
2
3 #taking input from user
4 sentence = input("Please give an input- ")
5 sentence=sentence.lower().strip()
6 i=0
7 #defining empty dictionary to use later to store
   element, counter pairs and eliminating common letters
   and words
8 dict={}
9
10 #checking if the string input is a word or a sentence
11 if " " not in sentence:
12     #searching for common elements
13     while i<len(sentence):
14         counter=0
15         k=0
16         while k<len(sentence):
17             if sentence[i]==sentence[k]:
18                 counter=counter+1
19                 k=k+1
20             else:
21                 k=k+1
22         #storing the values in dictionary
23         dict[f"{sentence[i]}"]=counter
24         i=i+1
25
26 else:
27     #making a list of words using split method
28     list = str.split(sentence)
29     #searching for common words
30     while i<len(list):
31         counter=0
32         k=0
33         while k<len(list):
34             if list[i]==list[k]:
35                 counter=counter+1
36                 k=k+1
37             else:
38                 k=k+1
39         #storing the pairs in dictionary

```

```

40         dict[f"{list[i]}"]=counter
41         i=i+1
42 #Printing the final result
43 print("Total occurrences")
44 for key,value in dict.items():
45     print(f"{key}-{value}")
46
47 *****#*****#*****#
48 ***#
49 #QUESTION-2
50
51 # Taking input from user
52 day = int(input('Please enter Day- '))
53 month = int(input('Please enter Month- '))
54 year = int(input('Please enter Year- '))
55
56 # Removing all the invalid cases
57 if day > 30 and month in {2, 4, 6, 9, 11}:
58     condition = False
59 elif day > 31 and month in {1, 3, 5, 7, 8, 10, 12}:
60     condition = False
61 elif (day == 29 or day == 30) and month == 2 and year
    % 4 != 0:
62     condition = False
63 elif day == 30 and month == 2 and year % 4 == 0:
64     condition = False
65 else:
66     condition = True
67
68 # After checking the condition, following if-else
statement is executed
69 if condition:
70     # checking and changing for new year
71     if day == 31 and month == 12:
72         n_year = year + 1
73     else:
74         n_year = year
75     if month == 12 and day == 31:
76         n_month = 1
77     else:

```

```

78         n_month = month
79         # changing dates
80         # checking for months with 31 days
81         if month in {1, 3, 5, 7, 8, 10, 12}:
82             if day == 31:
83                 next_day = 1
84                 if month != 12:
85                     n_month = month + 1
86                 else:
87                     n_month = 1
88             else:
89                 next_day = day + 1
90         # checking for the month of february
91         elif month == 2:
92             if year % 4 == 0:
93                 if day == 29:
94                     next_day = 1
95                     n_month = 3
96                 else:
97                     next_day = day + 1
98             else:
99                 if day == 28:
100                     next_day = 1
101                     n_month = 3
102                 else:
103                     next_day = day + 1
104
105         # covering all the remaining cases
106         else:
107             if day == 30:
108                 next_day = 1
109                 n_month = month + 1
110             else:
111                 next_day = day + 1
112         # printing the calculations
113         print(f"Next Date is: {next_day}/{n_month}/{
n_year}")
114     else:
115         # gives a warning and ends the program
116         print("That's not a valid date")
117

```

```

118 #####
    ***#
119
120 #QUESTION-3
121
122 l_range=int(input("Enter the lower range:"))
123 u_range=int(input("Enter the upper range:"))
124 a=[(x,x**2) for x in range(l_range,u_range+1)]
125 print(a)
126
127
128 #####
129
130 #QUESTION-4
131
132 grade=int(input('Enter Grade Points- '))
133
134 dict={10: {'grade_inp': 'A+', 'remarks': 'Outstanding'},
135       9: {'grade_inp': 'A', 'remarks': 'Excellent'
136       },
137       8: {'grade_inp': 'B+', 'remarks': 'Very Good'
138       },
139       7: {'grade_inp': 'B', 'remarks': 'Good'},
140       6: {'grade_inp': 'C+', 'remarks': 'Average'},
141       5: {'grade_inp': 'C', 'remarks': 'Below
    Average'},
142       4: {'grade_inp': 'D', 'remarks': 'Poor'}}
143 if 3<grade<11:
144     for key in dict.keys():
145         if key==grade:
146             value=dict[key]
147             print(f"Your Grade is {value['grade_inp']}
148             and {value['remarks']} performance ")
149         else:
150             continue
151 else:
152     print('Error!')
153
154 #####
    *****#

```

```

153
154 #QUESTION-5
155 alphabets = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H'
156             , 'I', 'J', 'K']
157
158 for row in range(0,6):
159     column=0
160     counter=0
161     while column<11:
162         if column<row or column>11-row-1:
163             print(" ", end="")
164         else:
165             print(alphabets[counter], end="")
166             counter=counter+1
167             column=column+1
168     print("")
169
170
171 *****#*****#*****#*****#*****#
172 *****#
173 #QUECTION-6
174
175 condition=True
176
177 #Defining dictionary to store the values
178 Dictionary={}
179 prompt="y"
180 while condition:
181     if prompt.lower()=="y":
182         SID=int(input("Please Enter SID of Student
183 - "))
184         name=input("Please enter the name of the
185 Student- ")
186         Dictionary[SID]=name
187         prompt= input("If you want to enter more
188 details press Y/N- ")
189         condition = True
190     else:

```

```

189         condition = False
190
191     print("Part a")
192     for key,value in Dictionary.items():
193         print(f"{key} is {value}")
194     print("")
195
196     print("Part b")
197     Val_sort_dict= sorted(Dictionary.values())
198     print(f"value sorted dictionary is {Val_sort_dict}")
199     print("")
200
201     print("Part c")
202     Key_sort_dict= sorted(Dictionary.keys())
203     print(f"Keys sorted dictionary is {Key_sort_dict}")
204     print("")
205
206     print("Part D")
207     SID_tbf=int(input("Please enter the student's SID
        whose detail you need- "))
208     if SID_tbf in Dictionary.keys():
209         print(f"Name of the required student is {
            Dictionary[SID_tbf]}")
210     else:
211         print("The SID is not present in the given Data"
            )
212     print("")
213
214
215     #####
216
217     #QUESTION-7
218
219     number=int(input("Total elements of Fibonacci
        sequence that you want(must be greater than 1)- "))
220
221     #using the formula of the sum of previous two terms
        is equal to the present term.
222     a_n1=1
223     a_n2=0

```

```

224 n=0
225 #initializing sum with first two terms
226 sum=a_n1+a_n2
227
228 #printing the initial two terms as the recursion is
    not valid on them
229 print(f"Fibonnaci sequence with {number} terms")
230 print(a_n2)
231 print(a_n1)
232
233 #Printing the remaining fibonnaci terms
234 while n<number-2:
235     a_n=a_n2+a_n1
236     a_n2=a_n1
237     a_n1=a_n
238     print(a_n)
239     n=n+1
240     sum+=a_n
241 average=sum/number
242 #printing the program end prompt
243 print(f"Average of total {number} terms of sequence
    is {average}")
244 print("END")
245
246
247 *****#*****#*****
    *****#
248
249 #QUESTION-8
250
251 set1= {1, 2, 3, 4, 5}
252 set2= {2, 4, 6, 8}
253 set3= {1, 5, 9, 13, 17}
254
255 #Q-(a)
256 set4=set1.union(set2)
257 print(set4)
258 set5=set1.intersection(set2)
259 print(set5)
260 #finally set
261 set=(set4)-(set5)

```

```
262 print(set)
263
264 #Q-(b)
265 set6=(set1).union(set2).union(set3)
266 print(set6)
267 set7=(set1).intersection(set2)
268 print(set7)
269 set8=(set2).intersection(set3)
270 print(set8)
271 set9=(set3).intersection(set1)
272 print(set9)
273 set10=(set7).union(set8).union(set9)
274 print(set10)
275 #finally set
276 set11=(set6)-(set10)
277 print(set11)
278
279 #Q-(c)
280 set7=(set1).intersection(set2)
281 print(set7)
282 set8=(set2).intersection(set3)
283 print(set8)
284 set9=(set3).intersection(set1)
285 print(set9)
286 set10=(set7).union(set8).union(set9)
287 #finally set#
288 #print(set10)
289
290 #Q-(d)
291 set11={1,2,3,4,5,6,7,8,9,10}
292 #finally set
293 set12=set11-set1
294 print(set12)
295
296 #Q-(e)
297 set13=(set1).union(set2).union(set3)
298 print(set13)
299 #finally set
300 set14=(set12)-(set13)
301 print(set14)
```