```python
#QUESTION-1

#taking input from user
sentence = input("Please give an input- ")
sentence=sentence.lower().strip()
i=0
#defining empty dictionary to use later to store element, counter pairs and eliminating common letters and words
dict={}

#checking if the string input is a word or a sentence
if " " not in sentence:
    #searching for common elements
    while i<len(sentence):
        counter=0
        k=0
        while k<len(sentence):
            if sentence[i]==sentence[k]:
                counter=counter+1
                k=k+1
            else:
                k=k+1
        #storing the values in dictionary
        dict[f"{sentence[i]}"]=counter
        i=i+1

else:
    #making a list of words using split method
    list = str.split(sentence)
    #searching for common words
    while i<len(list):
        counter=0
        k=0
        while k<len(list):
            if list[i]==list[k]:
                counter=counter+1
                k=k+1
            else:
                k=k+1
                #storing the pairs in dictionary
```

```
          dict[f"{list[i]}"]=counter
          i=i+1
#Printing the final result
print("Total occurances")
for key,value in dict.items():
    print(f"{key}-{value}")

#*****************#*********************#***************#

#QUESTION-2

# Taking input from user
day = int(input('Please enter Day- '))
month = int(input('Please enter Month- '))
year = int(input('Please enter Year- '))

# Removing all the invalid cases
if day > 30 and month in {2, 4, 6, 9, 11}:
    condition = False
elif day > 31 and month in {1, 3, 5, 7, 8, 10, 12}:
    condition = False
elif (day == 29 or day == 30) and month == 2 and year % 4 != 0:
    condition = False
elif day == 30 and month == 2 and year % 4 == 0:
    condition = False
else:
    condition = True

# After checking the condition, following if-else statement is executed
if condition:
    # checking and changing for new year
    if day == 31 and month == 12:
        n_year = year + 1
    else:
        n_year = year
    if month == 12 and day == 31:
        n_month = 1
    else:
```

```python
        n_month = month
    # changing dates
    # checking for months with 31 days
    if month in {1, 3, 5, 7, 8, 10, 12}:
        if day == 31:
            next_day = 1
            if month != 12:
                n_month = month + 1
            else:
                n_month = 1
        else:
            next_day = day + 1
    # checking for the month of february
    elif month == 2:
        if year % 4 == 0:
            if day == 29:
                next_day = 1
                n_month = 3
            else:
                next_day = day + 1
        else:
            if day == 28:
                next_day = 1
                n_month = 3
            else:
                next_day = day + 1

    # covering all the remaining cases
    else:
        if day == 30:
            next_day = 1
            n_month = month + 1
        else:
            next_day = day + 1
    # printing the calculations
    print(f"Next Date is: {next_day}/{n_month}/{n_year}")
else:
    # gives a warning and ends the program
    print("That's not a valid date")

```

```python
#*********************#***************#*****************#

#QUESTION-3

l_range=int(input("Enter the lower range:"))
u_range=int(input("Enter the upper range:"))
a=[(x,x**2) for x in range(l_range,u_range+1)]
print(a)


#**************#******************#*****************#

#QUESTION-4

grade=int(input('Enter Grade Points- '))

dict={10:{'grade_inp':'A+','remarks':'Outstanding'},
           9:{'grade_inp':'A','remarks':'Excellent'},
           8:{'grade_inp':'B+','remarks':'Very Good'},
           7:{'grade_inp':'B','remarks':'Good'},
           6:{'grade_inp':'C+','remarks':'Average'},
           5:{'grade_inp':'C','remarks':'Below Average'},
           4:{'grade_inp':'D','remarks':'Poor'}}
if 3<grade<11:
    for key in dict.keys():
        if key==grade:
            value=dict[key]
            print(f"Your Grade is {value['grade_inp']} and {value['remarks']} performance ")
        else:
            continue
else:
    print('Error!')


#*********************#*********************#***************#
```

```python

#QUESTION-5
alphabets = ['A', 'B', 'C', 'D', 'E', 'F', 'G', 'H', 'I', 'J', 'K']

for row in range(0,6):
    column=0
    counter=0
    while column<11:
        if column<row or column>11-row-1:
            print(" ", end="")

        else:
            print(alphabets[counter], end="")
            counter=counter+1
        column=column+1
    print("")


#******************#******************#******************#

#QUECTION-6

condition=True

#Defining dictionary to store the values
Dictionary={}
prompt="y"
while condition:
    if prompt.lower()=="y":
        SID=int(input("Please Enter SID of Student - "))
        name=input("Please enter the name of the Student- ")
        Dictionary[SID]=name
        prompt= input("If you want to enter more details press Y/N- ")
        condition = True

    else:
        condition = False

print("Part a")
for key,value in Dictionary.items():
    print(f"{key} is {value}")
print("")

print("Part b")
Val_sort_dict= sorted(Dictionary.values())
print(f"value sorted dictionary is {Val_sort_dict}")
print("")

print("Part c")
Key_sort_dict= sorted(Dictionary.keys())
print(f"Keys sorted dictionary is {Key_sort_dict}")
print("")

print("Part D")
SID_tbf=int(input("Please enter the student's SID whose detail you need- "))
if SID_tbf in Dictionary.keys():
    print(f"Name of the required student is {Dictionary[SID_tbf]}")
else:
    print("The SID is not present in the given Data")
print("")


#**********************#********************#***************#

#QUESTION-7

number=int(input("Total elements of Fibonacci sequence that you want(must be greater than 1)- "))

#using the formula of the sum of previous two terms is equal to the present term.
a_n1=1
a_n2=0
```

```python
n=0
#initializing sum with first two terms
sum=a_n1+a_n2

#printing the initial two terms as the recursion is not valid on them
print(f"Fibonnaci sequence with {number} terms")
print(a_n2)
print(a_n1)

#Printing the remaining fibonnaci terms
while n<number-2:
    a_n=a_n2+a_n1
    a_n2=a_n1
    a_n1=a_n
    print(a_n)
    n=n+1
    sum+=a_n
average=sum/number
#printing the program end prompt
print(f"Average of total {number} terms of sequence is {average}")
print("END")


#*********************#*******************#********************#

#QUESTION-8

set1= {1, 2, 3, 4, 5}
set2= {2, 4, 6, 8}
set3= {1, 5, 9, 13, 17}

#Q-(a)
set4=set1.union(set2)
print(set4)
set5=set1.intersection(set2)
print(set5)
#finally set
set=(set4)-(set5)
```

```python
print(set)

#Q-(b)
set6=(set1).union(set2).union(set3)
print(set6)
set7=(set1).intersection(set2)
print(set7)
set8=(set2).intersection(set3)
print(set8)
set9=(set3).intersection(set1)
print(set9)
set10=(set7).union(set8).union(set9)
print(set10)
#finally set
set11=(set6)-(set10)
print(set11)

#Q-(c)
set7=(set1).intersection(set2)
print(set7)
set8=(set2).intersection(set3)
print(set8)
set9=(set3).intersection(set1)
print(set9)
set10=(set7).union(set8).union(set9)
#finally set#
#print(set10)

#Q-(d)
set11={1,2,3,4,5,6,7,8,9,10}
#finally set
set12=set11-set1
print(set12)

#Q-(e)
set13=(set1).union(set2).union(set3)
print(set13)
#finally set
set14=(set12)-(set13)
print(set14)
```