

ACTIVITY 1.34

Q1. Write a python program to sort a list of 'n' integers using insertion sorting. Print the list of numbers Before and After sorting.

#code screenshot

The screenshot displays the Visual Studio Code interface. The Explorer panel on the left shows a project structure with files like 1.py, 2.py, pcc.py, and q1.py. The main editor window shows the code for q1.py, which implements insertion sorting. The code prompts the user for the limit of numbers and then for each number, appends them to a list, and prints the list before and after sorting. A video call window is overlaid on the right side of the editor, showing a person wearing a red shirt and a black beanie. The Windows taskbar at the bottom shows the system clock as 13:55 on 17-01-2021.

```
1 #sorting n integers using insertion sort
2 numberlist=[]
3 lim=int(input("Enter the limit of numbers you want in the list : "))
4 for i in range(1,lim+1):
5     num=int(input("Enter number {} : ".format(i)))
6     numberlist.append(num)
7 print("Number List Before Sorting : ",numberlist)
8 print("\n")
9 for i in range(1,len(numberlist)):
10     temp=numberlist[i]
11     j=i-1
12     while j>=0:
13         if temp<numberlist[j]:
14             numberlist[j+1]=numberlist[j]
15             numberlist[j]=temp
16         j=j-1
17 print("Number List After Sorting : ",numberlist)
18 print("\n")
```

AKSHAT KUMAR

20MIS0183

#code

```
#sorting n integers using insertion sort
numberlist=[]
lim=int(input("Enter the limit of numbers you want in the list : "))
for i in range(1,lim+1):
    num=int(input("Enter number {} : ".format(i)))
    numberlist.append(num)
print("Number List Before Sorting : ",numberlist)
print("\n")
for i in range(1,len(numberlist)):
    temp=numberlist[i]
    j=i-1
    while j>=0:
        if temp<numberlist[j]:
            numberlist[j+1]=numberlist[j]
            numberlist[j]=temp
        j=j-1
print("Number List After Sorting : ",numberlist)
print("\n")
```

AKSHAT KUMAR

20MIS0183

#output screenshot

The screenshot displays the Visual Studio Code interface with a Python file named `q1.py` open. The code implements an insertion sort algorithm. The terminal window shows the execution of the script, which prompts the user to enter the limit of numbers (10) and then enters 10 numbers: -2, 3, 400, -876, 3387, -43, 90, -12, 888, and -4. The output shows the list before and after sorting.

```
act1.34 > q1.py > ...
1 #sorting n integers using insertion sort
2 numberlist=[]
3 lim=int(input("Enter the limit of numbers you want in the list
4 for i in range(1,lim+1):
5     num=int(input("Enter number {} : ".format(i)))
6     numberlist.append(num)
7 print("Number List Before Sorting : ",numberlist)
8 print("\n")

Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.

Try the new cross-platform PowerShell https://aka.ms/pscore6

PS C:\Users\AKSHAT\Desktop\python> & C:/Users/AKSHAT/AppData/Local/Programs/Python/Python39-6/python.exe q1.py
Enter the limit of numbers you want in the list : 10
Enter number 1 : -2
Enter number 2 : 3
Enter number 3 : 400
Enter number 4 : -876
Enter number 5 : 3387
Enter number 6 : -43
Enter number 7 : 90
Enter number 8 : -12
Enter number 9 : 888
Enter number 10 : -4
Number List Before Sorting : [-2, 3, 400, -876, 3387, -43, 90, -12, 888, -4]

Number List After Sorting : [-876, -43, -12, -4, -2, 3, 90, 400, 888, 3387]

PS C:\Users\AKSHAT\Desktop\python>
```

A video call window titled "Camera" is overlaid on the right side of the editor, showing a person wearing a red shirt and a black beanie. The person's face is framed by a blue rectangle, indicating face tracking or focus. The video call interface includes standard controls like a camera icon and a close button.

AKSHAT KUMAR

20MIS0183

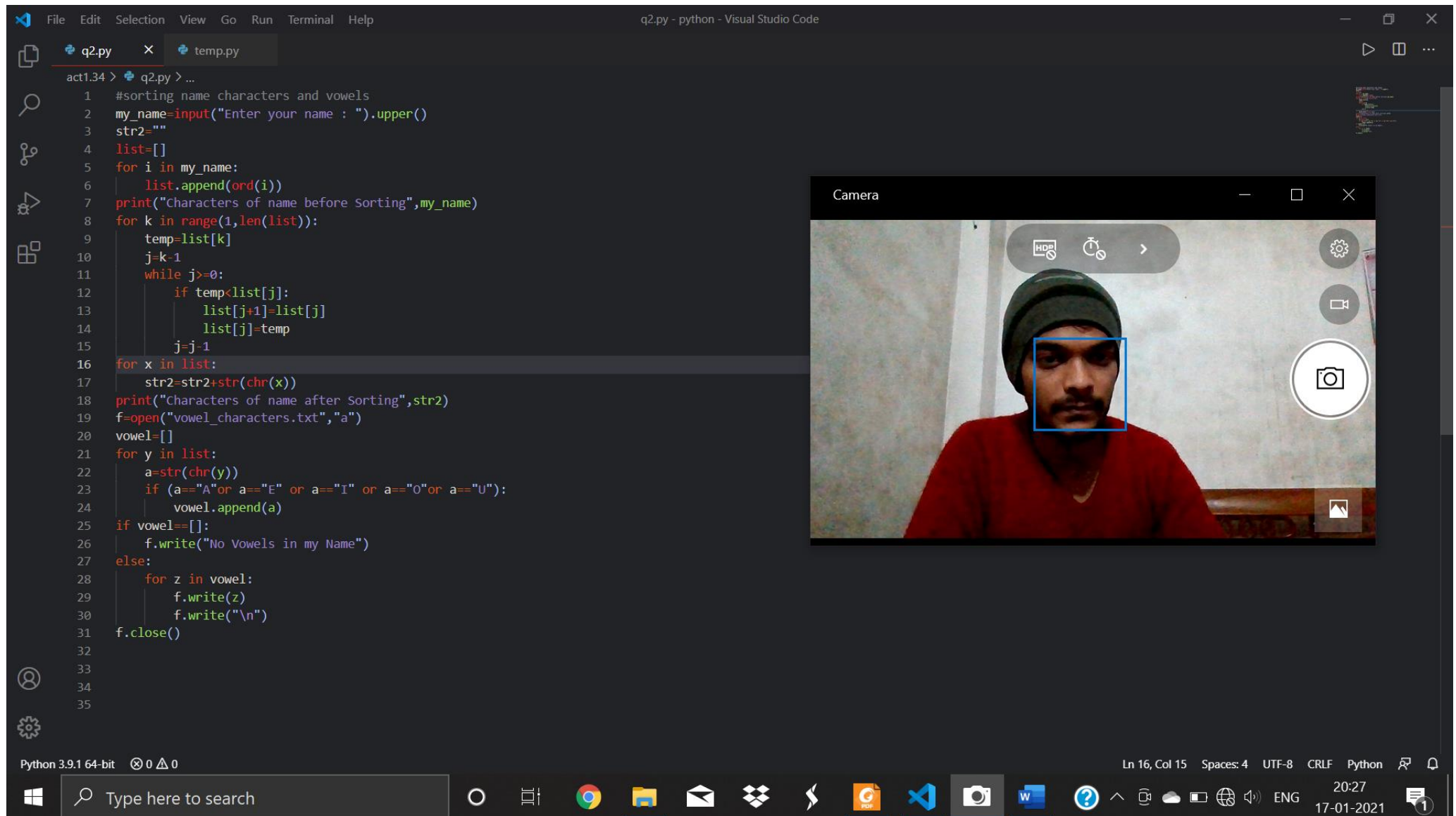
Q2. Write a python program for the following requirements.

- a) Read your name using input () and store it in a string variable (say my_name).*
- b) Print my_name on the screen.*
- c) Using insertion sorting technique, sort the characters in my_name and print them in ascending order.*
- d) Select only the vowel characters from my_name and store them into a file (say 'vowel_characters.txt'). If your name does not contain any vowel character, then store 'NO VOWELS IN MY NAME' into 'vowel_characters.txt'.*
- e) Trace the program manually (handwritten material) in neat format to show the status of all variables and condition checking for "first five characters of your name". Add the scanned soft copy into the answer sheet (PDF file).*

AKSHAT KUMAR

20MIS0183

#code screenshot



AKSHAT KUMAR

20MIS0183

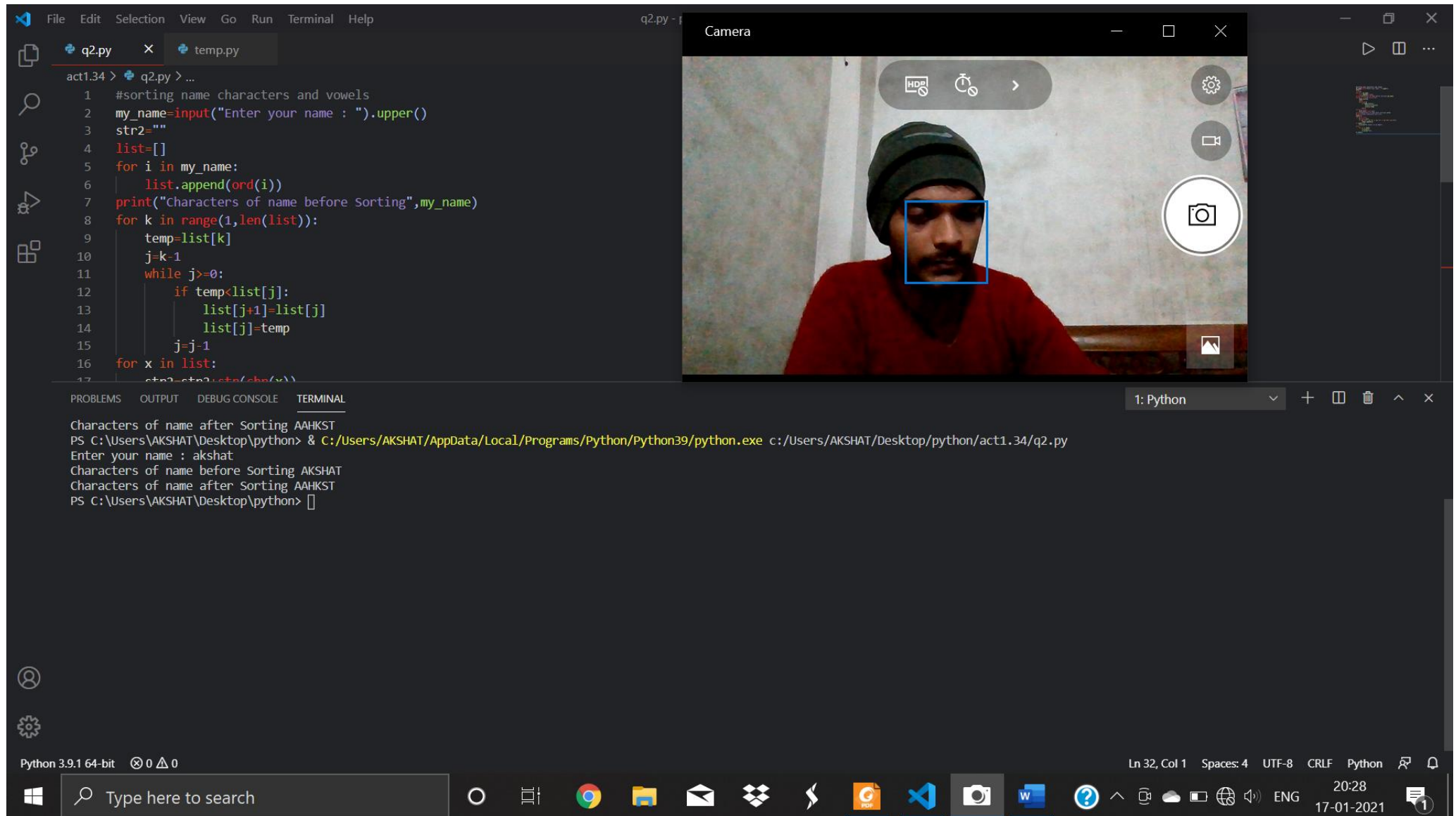
#code

```
#sorting name characters and vowels
my_name=input("Enter your name : ").upper()
str2=""
list=[]
for i in my_name:
    list.append(ord(i))
print("Characters of name before Sorting",my_name)
for k in range(1,len(list)):
    temp=list[k]
    j=k-1
    while j>=0:
        if temp<list[j]:
            list[j+1]=list[j]
            list[j]=temp
        j=j-1
for x in list:
    str2=str2+str(chr(x))
print("Characters of name after Sorting",str2)
f=open("vowel_characters.txt","a")
vowel=[]
for y in list:
    a=str(chr(y))
    if (a=="A" or a=="E" or a=="I" or a=="O" or a=="U"):
        vowel.append(a)
if vowel==[]:
    f.write("No Vowels in my Name")
else:
    for z in vowel:
        f.write(z)
        f.write("\n")
f.close()
```


AKSHAT KUMAR

20MIS0183

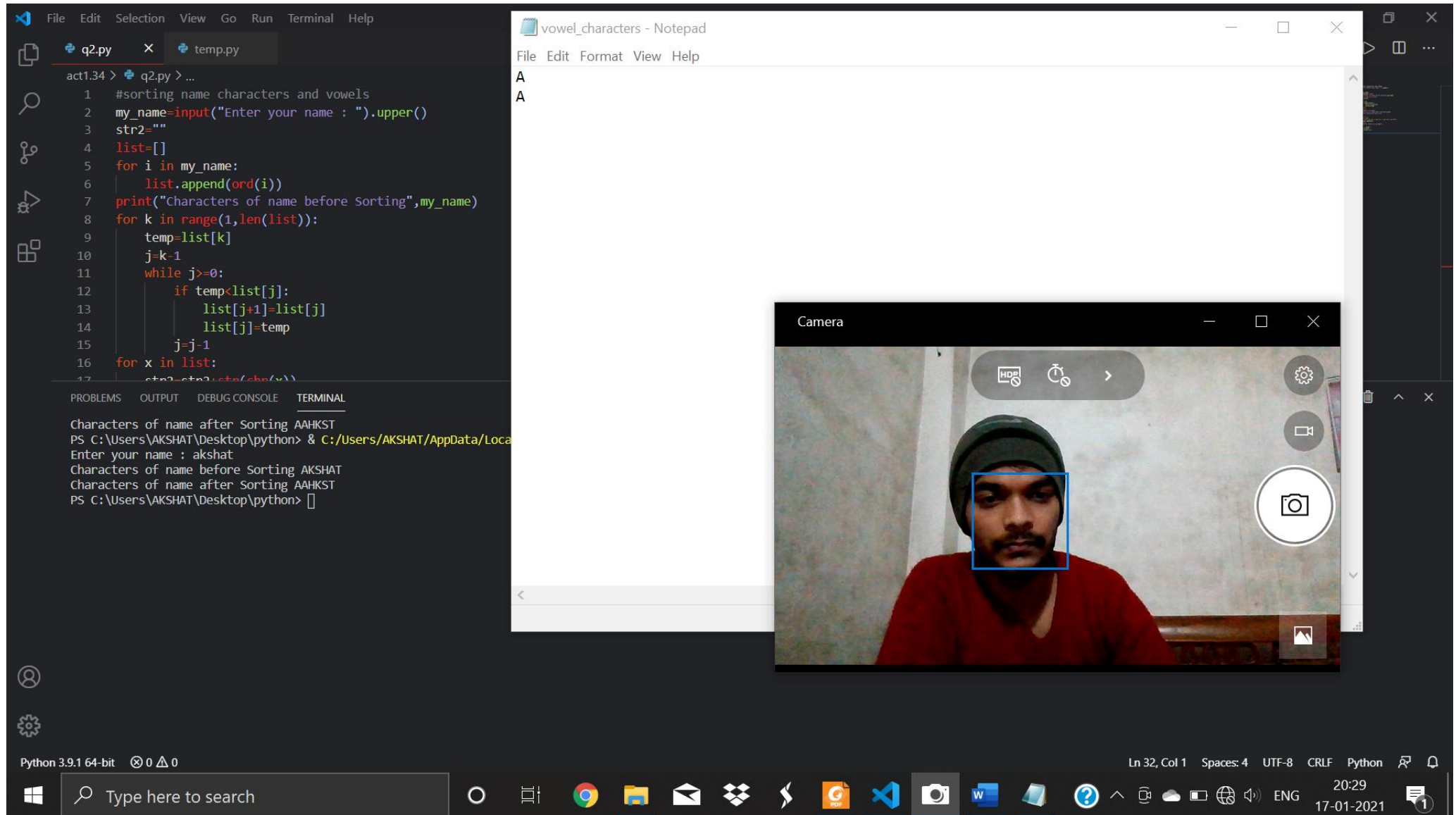
#output screenshot



AKSHAT KUMAR

20MIS0183

#file created



#tracing

```

my_
>> name = input("Enter your name: ").upper()  → INPUT
>> str1 = " "                                → OUTPUT
>> list = []                                AKSHAT
>> for i in my_ name: (ie i = A, K, S, H, A, T)
    list.append(ord(i))
        i.e. [65, 75, 83, 72, 65, 84]
            0 1 2 3 4 5
my name characters are stored in list as ascii codes
>> print(my_name)    i.e. AKSHAT    (Before sorting)
>> for k in range(1, len(list)):
    len(list) = 6
    >> for k in range(1, 6): (i.e. 1, 2, 3, 4, 5)
        >> k = 1
        >> temp = list[k]
        >> temp = list[1]
        >> temp = 75
        j = k-1 = 1-1 = 0
        while j >= 0:
            i.e. true.
            if 75 < 65 : True False
            no swapping    list = [65, 75, 83, 72, 65, 84]
                                0 1 2 3 4 5
        >> k = 2
        >> temp = 83
        j = 1
        while j >= 0: True
            if 83 < 75 : False
            no swapping    list = [65, 75, 83, 72, 65, 84]
                                0 1 2 3 4 5
        j = 0
        while j >= 0: True
            if 83 < 65 : False
            no swapping    list = [65, 75, 83, 72, 65, 84]
                                0 1 2 3 4 5

```

①

```

>> k = 3
temp = 72
j = 2
while j >= 0: True
    if 72 < 83 : True
        72 ↔ 83
        swapping takes place    list = [65, 75, 72, 83, 65, 84]
                                0 1 2 3 4 5
    j = 1
    while j >= 0: True
        if 72 < 75 : True
            72 ↔ 75
            swapping takes place    list = [65, 72, 75, 83, 65, 84]
                                    0 1 2 3 4 5
        j = 0
        while j >= 0: True
            if 72 < 65 : False
            no swapping
            list = [65, 72, 75, 83, 65, 84]
                                0 1 2 3 4 5
>> k = 4
temp = 65
j = 3
while j >= 0: True
    if 65 < 83 : True
        65 ↔ 83
        swapping takes place    list = [65, 72, 75, 65, 83, 84]
                                0 1 2 3 4 5
    j = 2
    while j >= 0: True
        if 65 < 75 : True
            65 ↔ 75
            swapping takes place    list = [65, 72, 65, 75, 83, 84]
                                    0 1 2 3 4 5
    j = 1
    while j >= 0: True
        if 65 < 72 : True
            65 ↔ 72
            swapping takes place    list = [65, 65, 72, 75, 83, 84]
                                    0 1 2 3 4 5

```

②

```
j = 0
while j >= 0: True
if 65 < 65: False
no swapping
```

list = [65, 65, 72, 75, 83, 84]
0 1 2 3 4 5

>> k = 5

```
temp = 84
j = 4
while j >= 0: true
if 84 < 83: False
no swapping
```

list = [65, 65, 72, 75, 83, 84]

```
j = 3
while j >= 0: true
if 84 < 75: False
no swapping
```

list = [65, 65, 72, 75, 83, 84]

```
j = 2
while j >= 0: true
if 84 < 72: False
no swapping
```

list = [65, 65, 72, 75, 83, 84]

```
j = 1
while j >= 0: true
if 84 < 65: False
no swapping
```

list = [65, 65, 72, 75, 83, 84]

```
j = 0
while j >= 0: true
if 84 < 65: False
no swapping
```

list = [65, 65, 72, 75, 83, 84]

Final list = [65, 65, 72, 75, 83, 84] (name after sorting)
for x in list:
str2 = str2 + str(chr(x))
str2 = chr(65) + chr(65) + chr(72) + chr(75) + chr(83) + chr(84)
AAHKST