Question 1 :

Assuming that we have some email addresses in the "username@companyname.com" format, please write a program to print the company name of a given email address. Both user names and company names are composed of letters only

In [13]:
```
str3=input("enter the your mail" )
k=str3.find("@")
i=k+1
while(i>k):
    if str3[i]==".":
        break
    print(str3[i],end="")
    i=i+1
```

```
enter the your mailram@amazon.com
amazon
```

Question 2 :

Write a program that accepts a comma-separated sequence of words as input and prints the words in a comma-separated sequence after sorting them alphabetically

In [12]:
```
str6=input("Enter comma seperated words : ")
x=str6.split(",")
x.sort()
print(x)
```

```
Enter comma seperated words : b,d,e,a
['a', 'b', 'd', 'e']
```

Question 3:

Create your own Jupyter Notebook for Sets

SETS

sets are written inside curly brackets.

They are not ordered and indexing is not present in the sets

set1={"a","b","c"}

as sets are unordered so we can not know how the elements of sets are printed while printing

We can not access the elements in the set using index,because they are not ordered

But we can access the elements in sets by using the for loop or while loop

In [2]:
```
set1={"a","b","c"}
for i in set1:
    print(i)
```

```
c
a
b
```

In [3]:
```
print("a" in set1)
```

```
True
```

once the set is created we cannot change the elements present in the list but we can add or update the set, but we can't delete the elements of the set

In [4]:
```
set1.add('e')
print(set1)
```

```
{'c', 'a', 'b', 'e'}
```

In [5]:
```
set1.update('e','f','g','h')
print(set1)
```

```
{'h', 'a', 'g', 'e', 'f', 'c', 'b'}
```

If add an element that is already present in the set then, we cannot have same values in the set

In [6]:
```
print(len(set1))
```

```
7
```

We can also remove or delete or dicard an element from the set by using the functions remove() and discard().

the diffrence between two of them is :

remove() function removes the element that is present in the set and if the element is not present in the set then it returns the error

It is same with the discard,but if the element is not presnt then it does not give an error

In [11]:
```
set1.remove("f")
print(set1)
```

```
{'h', 'a', 'e', 'c', 'b'}
```

In [12]:
```
set1.discard('e')
print(set1)
```

```
{'h', 'a', 'c', 'b'}
```

In [13]:
```
set1.remove("z");
```

```
---------------------------------------------------------------------------
KeyError                                  Traceback (most recent call last)
<ipython-input-13-e6c05b7950c1> in <module>
----> 1 set1.remove("z");

KeyError: 'z'
```

In [15]:
```
set1.discard('z')
```

we can also use pop method but we do not know which element as been deleted

In [16]:
```
set1.pop()
```

Out[16]:
```
'h'
```

In [17]:
```
print(set1)
```

```
{'a', 'c', 'b'}
```

To clear all elements in the set we use clear()

In [18]:
```
set1.clear()
print(set1)
```

```
set()
```

We can also delete the set using the delete method

In [19]:
```
set2={1,2,3,4,5,6}
del set2
print(set2)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-19-dce6f5ecce25> in <module>
      1 set2={1,2,3,4,5,6}
      2 del set2
----> 3 print(set2)

NameError: name 'set2' is not defined
```

we can join two sets using the union method

using union method we are combining all the elements from set1 and set2 into a new set ie set3

we can update the two sets ie insert all elements of the set2 into set1 using update method

In [21]:
```
set1={1,2,3,4,5}
set2={6,7,8,9,10}
set2.update(set1)
print(set2)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

because we can not have or we cannot insert same elements in the sets concept.

both union and update will exclude the common or duplicate elements

In [22]:
```
set3=set2.union(set1)
print(set3)
```

```
{1, 2, 3, 4, 5, 6, 7, 8, 9, 10}
```

Question 4:

Given a list of n-1 numbers ranging from 1 to n, your task is to find the missing number. There are no

In [4]:
```
def find_missing(lst):
    return [x for x in range(lst[0], lst[-1]+1)
                            if x not in lst]


lst = [1, 2, 4, 6, 7, 9, 10]
print(find_missing(lst))
```

```
[3, 5, 8]
```

In [9]:
```
list4=[1,2,2,3,4,5]
for i in list4:
    for j in list4:
        if i==j:
            list4.remove(j);
print(list3)
```