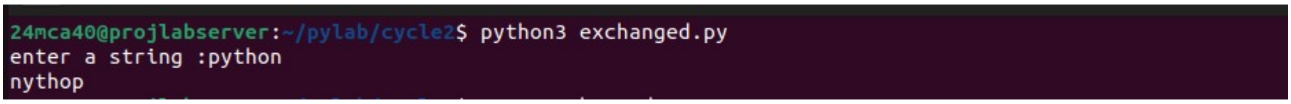
PROGRAM1

AIM
create a string from the given string where the first and last chracters are exchanged

SOURCE CODE

```
string=input("enter a string :")
newstring=string[-1]+string[1:-1]+string[0]
print(newstring)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 exchanged.py
enter a string :python
nythop
```

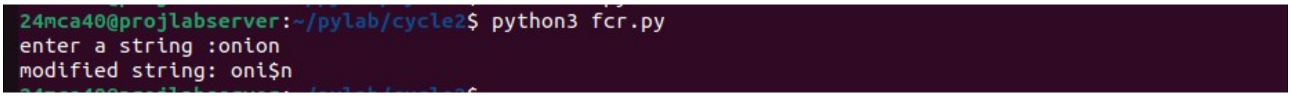
PROGRAM2

AIM
Get a string from an input string where all occurences of the first character are replaced with ‘$’, except the first character

SOURCE CODE

```
tring=input("enter a string :")
first_char=string[0]
new_string=first_char+string[1:].replace(first_char,"$")
print("modified string:",new_string)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 fcr.py
enter a string :onion
modified string: oni$n
```

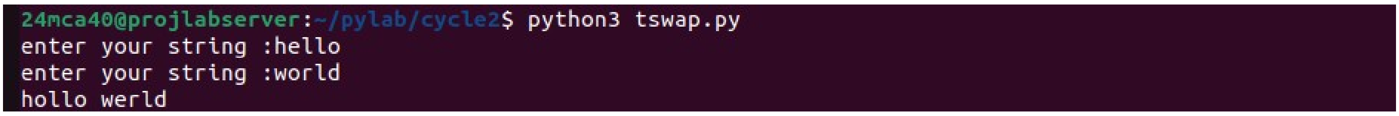
PROGRAM3

AIM
create a single string seperated with space from two strings by swapping the characters at position1

SOURCE CODE

```
string=input("enter a string :")
first_char=string[0]
new_string=first_char+string[1:].replace(first_char,"$")
print("modified string:",new_string)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 tswap.py
enter your string :hello
enter your string :world
hollo werld
```

PROGRAM4

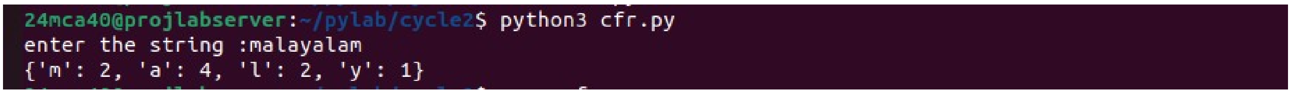
AIM

count the number of charcters(character frequency) in a string

SOURCE CODE

```
n=input("enter the string :").lower()
s={}
for i in n:
        if i in s:
                s[i]+=1
        else:
                s[i]=1
print(s)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 cfr.py
enter the string :malayalam
{'m': 2, 'a': 4, 'l': 2, 'y': 1}
```

PROGRAM5

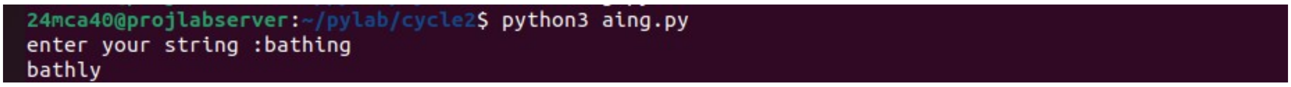
AIM

adding ‘ing’ at the end of the given string.if it already ends with ‘ing’,then add ‘ily’

SOURCE CODE

```
string=input("enter your string :")
if string[-3:]=="ing":
        print(string[:-3]+"ly")
else:
        print(string+"ing")
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 aing.py
enter your string :bathing
bathly
```

PROGRAM6

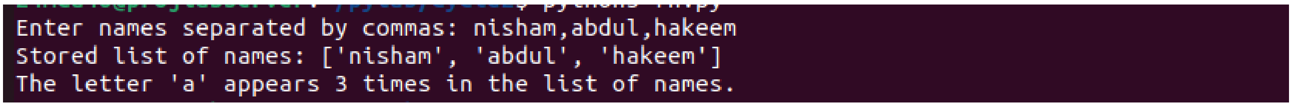
AIM

Store a list of first names.count the occurence of ‘a’ within the list

SOURCE CODE

```
names = input("Enter names separated by commas: ").split(',')
names = [name.strip() for name in names]
print("Stored list of names:", names)
count_a = sum(name.strip().lower().count('a') for name in names)
print(f"The letter 'a' appears {count_a} times in the list of names.")
```

OUTPUT



```
Enter names separated by commas: nisham,abdul,hakeem
Stored list of names: ['nisham', 'abdul', 'hakeem']
The letter 'a' appears 3 times in the list of names.
```

PROGRAM 7

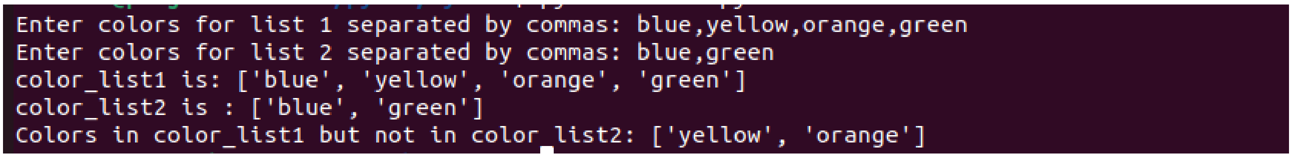
AIM

write a python programme for read two lists color list1 and list2 print an colors from color_list1 not be contained in colorlist_2

```
Scolor_list1 = input("Enter colors for list 1 separated by commas: ").split(',')
color_list2 = input("Enter colors for list 2 separated by commas: ").split(',')


color_list1 = [color.strip() for color in color_list1]
color_list2 = [color.strip() for color in color_list2]
print("color_list1 is:",color_list1)
print("color_list2 is :",color_list2)
unique_colors = [color for color in color_list1 if color not in color_list2]
print("Colors in color_list1 but not in color_list2:", unique_colors)OURCE CODE
```

OUTPUT



```
Enter colors for list 1 separated by commas: blue,yellow,orange,green
Enter colors for list 2 separated by commas: blue,green
color_list1 is: ['blue', 'yellow', 'orange', 'green']
color_list2 is : ['blue', 'green']
Colors in color_list1 but not in color_list2: ['yellow', 'orange']
```

PROGRAM8

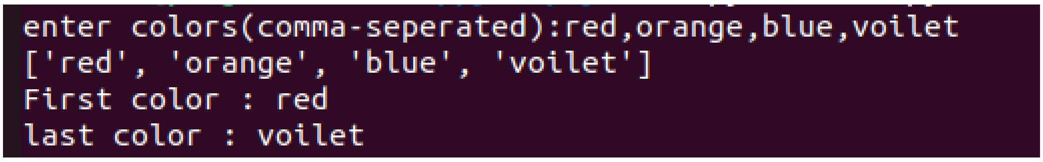
AIM

create a list of colors from comma-seperated color names entered by the user.Display first and last colors

SOURCE CODE

```
olors=input("enter colors(comma-seperated):").split(',')
colors=[color.strip()for color in colors]
print(colors)
print("First color :",colors[0])
print("last color :",colors[-1])
```

OUTPUT



```
enter colors(comma-seperated):red,orange,blue,voilet
['red', 'orange', 'blue', 'voilet']
First color : red
last color : voilet
```

PROGRAM 9

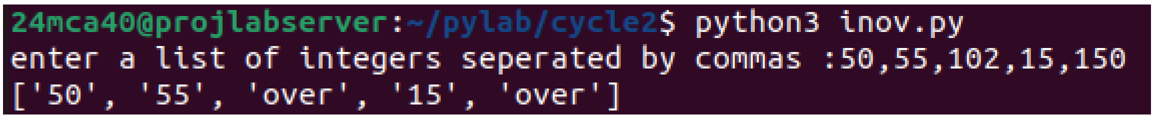
AIM

write a program to prompt the user for a list of integers for on values greater than 100,store ‘over’ instead

SOURCE CODE

```
integers=input("enter a list of integers seperated by commas :").split(',')
for i in range(len(integers)):
        if int(integers[i])>100:
                integers[i]='over'
print(integers)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 inov.py
enter a list of integers seperated by commas :50,55,102,15,150
['50', '55', 'over', '15', 'over']
```

PROGRAM 10

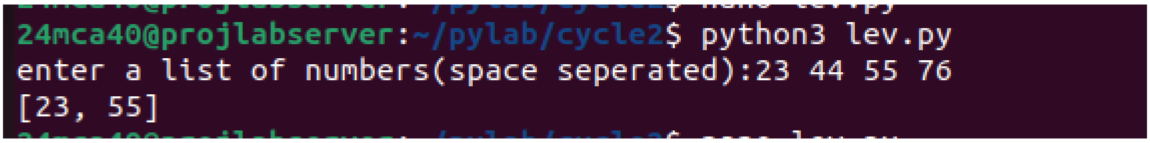
AIM

From a list of integers,create a list after removing even numbers

SOURCE CODE

```
lst=[int(num) for num in input("enter a list of numbers(space seperated):").split()]
odd_lst=[odd for odd in lst if odd%2!=0]
print(odd_lst)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 lev.py
enter a list of numbers(space seperated):23 44 55 76
[23, 55]
24mca40@projlabserver:~/pylab/cycle2$ nano lev.py
```

PROGRAM 11

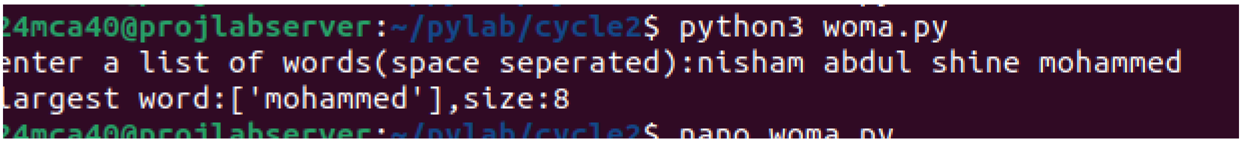
AIM

Accept a list of words and return the length of the longest word.

SOURCE CODE

```
lst=input("enter a list of words(space seperated):").split()
maxlength=max(len(word) for word in lst)
lg_word=[word for word in lst if len(word)==maxlength]
print(f"largest word:{lg_word},size:{maxlength}")
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 woma.py
enter a list of words(space seperated):nisham abdul shine mohammed
largest word:['mohammed'],size:8
24mca40@projlabserver:~/pylab/cycle2$ nano woma.py
```

PROGRAM 12

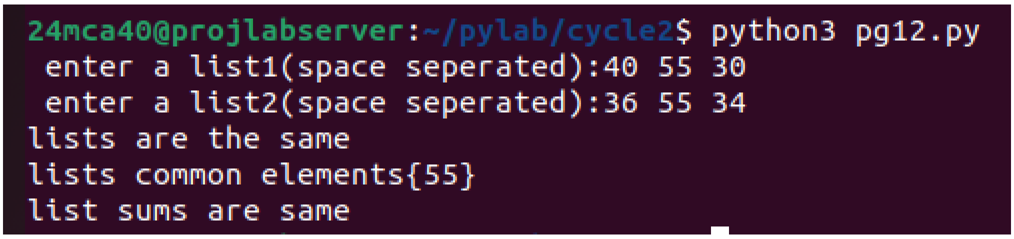
AIM

Write a program to prompt the user to enter two lists of integers and check
(a) Whether lists are of the same length.
(b) Whether the list sums to the same value.
(c) Whether any value occurs in both Lists

SOURCE CODE

```
lst1=[int(num) for num in input(" enter a list1(space seperated):").split()]
lst2=[int(num) for num in input(" enter a list2(space seperated):").split()]
length=len(lst1)==len(lst2)
lsum=sum(lst1)==sum(lst2)
common=set(lst1)&set(lst2)
if length:
        print("lists are the same")
else:
        print("the lists are not same")
print(f"lists common elements{common}")
if lsum:
        print(f"list sums are same")
else:
        print("the list sums are not same")
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 pg12.py
 enter a list1(space seperated):40 55 30
 enter a list2(space seperated):36 55 34
lists are the same
lists common elements{55}
list sums are same
```

PROGRAM13

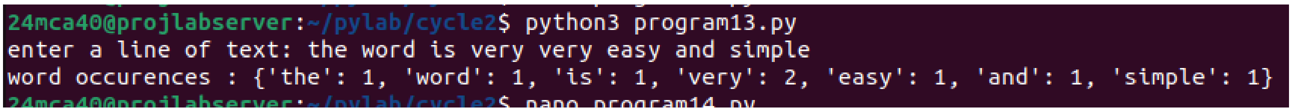
AIM

program to count the occurance of each word in a line of text

SOURCE CODE

```
ext=input("enter a line of text: ")
words=text.split()
word_count={}
for word in words:
        word=word.lower()
        if word in word_count:
                word_count[word]+=1
        else:
                word_count[word]=1
print("word occurences :",word_count)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 program13.py
enter a line of text: the word is very very easy and simple
word occurences : {'the': 1, 'word': 1, 'is': 1, 'very': 2, 'easy': 1, 'and': 1, 'simple': 1}
24mca40@projlabserver:~/pylab/cycle2$ nano program14.py
```

PROGRAM14

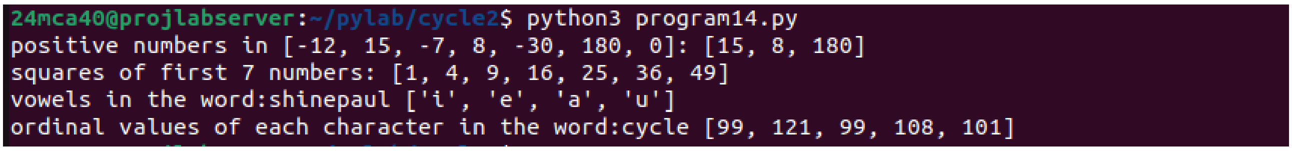
AIM

List comprehensions:
(a) Generate positive list of numbers from a given list of integers
(b) Square of N numbers
(c) Form a list of vowels selected from a given word
(d) Form a list ordinal value of each element of a word (Hint: use ord() to get ordinal values)
Sort dictionary in ascending and descending order.
Merge two dictionaries.

SOURCE CODE

```
numbers=[-12,15,-7,8,-30,180,0]
pos=[num for num in numbers if num>0]
print(f"positive numbers in {numbers}:",pos)
N=7
sqr=[num**2 for num in range(1,N+1)]
print("squares of first 7 numbers:",sqr)
word="shinepaul"
vowels=[char for char in word if char in 'aeiouAEIOU']
print(f"vowels in the word:{word}",vowels)
word="cycle"
ordinal_values=[ord(char) for char in word]
print("ordinal values of each character in the word:cycle",ordinal_values)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 program14.py
positive numbers in [-12, 15, -7, 8, -30, 180, 0]: [15, 8, 180]
squares of first 7 numbers: [1, 4, 9, 16, 25, 36, 49]
vowels in the word:shinepaul ['i', 'e', 'a', 'u']
ordinal values of each character in the word:cycle [99, 121, 99, 108, 101]
```

PROGRAM15

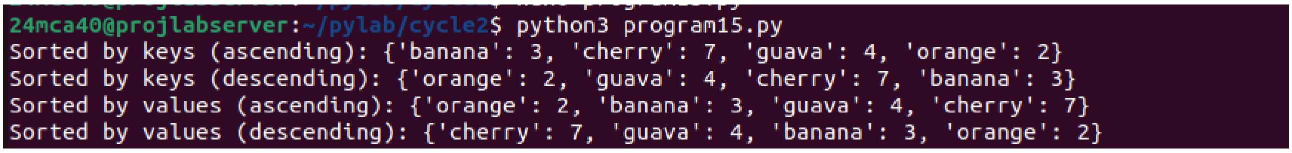
AIM

Sort dictionary in ascending and descending order.

SOURCE CODE

```
my_dict = {'banana': 3, 'cherry': 7, 'orange': 2, 'guava': 4}
keys_asc = dict(sorted(my_dict.items()))
print("Sorted by keys (ascending):", keys_asc)
keys_desc = dict(sorted(my_dict.items(), reverse=True))
print("Sorted by keys (descending):", keys_desc)
values_asc = dict(sorted(my_dict.items(), key=lambda item: item[1]))
print("Sorted by values (ascending):", values_asc)
values_desc = dict(sorted(my_dict.items(), key=lambda item: item[1], reverse=True))
print("Sorted by values (descending):", values_desc)
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 program15.py
Sorted by keys (ascending): {'banana': 3, 'cherry': 7, 'guava': 4, 'orange': 2}
Sorted by keys (descending): {'orange': 2, 'guava': 4, 'cherry': 7, 'banana': 3}
Sorted by values (ascending): {'orange': 2, 'banana': 3, 'guava': 4, 'cherry': 7}
Sorted by values (descending): {'cherry': 7, 'guava': 4, 'banana': 3, 'orange': 2}
```

PROGRAM16

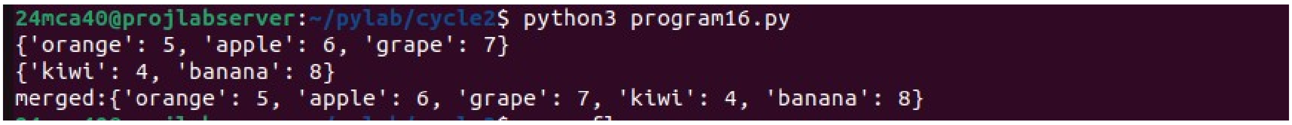
AIM

program to merge two dictionaries

SOURCE CODE

```
dict1={'orange':5,'apple':6,'grape':7}
dict2={'kiwi':4,'banana':8}
print(dict1)
print(dict2)
dict1.update(dict2)
print(f"merged:{dict1}")
```

OUTPUT



```
24mca40@projlabserver:~/pylab/cycle2$ python3 program16.py
{'orange': 5, 'apple': 6, 'grape': 7}
{'kiwi': 4, 'banana': 8}
merged:{'orange': 5, 'apple': 6, 'grape': 7, 'kiwi': 4, 'banana': 8}
```