

# **FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)<sup>TM</sup>**

**HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577**



**FOCUS ON EXCELLENCE**

**20MCA131 PROGRAMMING LAB**

**LABORATORY RECORD**

**Name: ABIMA YUGESH M**

**Branch: MASTER OF COMPUTER APPLICATIONS**

**Semester: 1      Batch: A      Roll No: 05**

**MARCH 2022**

# FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY (FISAT)<sup>TM</sup>

HORMIS NAGAR, MOOKKANNOOR, ANGAMALY-683577



FOCUS ON EXCELLENCE

## CERTIFICATE

*This is to certify that this is a Bonafide record of the Practical work done by ABIMA YUGESH M in the 20MCA131 PROGRAMMING LAB Laboratory towards the partial fulfilment for the award of the Master Of Computer Applications during the academic year 2021-2022.*

Signature of Staff in Charge

Name:

Signature of H O D

Name:

**Date of University practical examination .....**

Signature of  
Internal Examiner

Signature of  
External Examiner

**CONTENT**

<b>SI No:</b>	<b>Date :</b>	<b>Name of Experiment:</b>	<b>Page No:</b>	<b>Signature of Staff –In – Charge:</b>
<b>1</b>		Display future leap years from current year to a final year entered by user.	<b>5</b>	
<b>2</b>		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) List ordinal value of each element of a word (Hint: use ord() to get ordinal values)	<b>5</b>	
<b>3</b>		Count the occurrences of each word in a line of text.	<b>8</b>	
<b>4</b>		Prompt the user for a list of integers. For all values greater than 100, store 'over' instead	<b>9</b>	
<b>5</b>		Store a list of first names. Count the occurrences of 'a' within the list	<b>10</b>	
<b>6</b>		Enter 2 lists of integers. Check (a) Whether list are of same length (b) whether list sums to same value (c) whether any value occur in both.	<b>10</b>	
<b>7</b>		Get a string from an input string where all occurrences of first character replaced with '\$', except first character	<b>12</b>	
<b>8</b>		Create a string from given string where first and last characters exchanged. [eg: python -> nythop]	<b>13</b>	
<b>9</b>		Accept the radius from user and find area of circle.	<b>13</b>	
<b>10</b>		Find biggest of 3 numbers entered.	<b>14</b>	
<b>11</b>		Accept a file name from user and print extension of that.	<b>14</b>	

12		Create a list of colors from comma-separated color names entered by user. Display first and last colors.	15	
13		Accept an integer n and compute $n+nn+nnn$ .	16	
14		Print out all colors from color-list1 not contained in color-list2.	16	
15		Create a single string separated with space from two strings by swapping the character at position 1.	17	
16		Merge two dictionaries.	18	
17		Find gcd of 2 numbers.	19	
18		From a list of integers, create a list removing even numbers.	20	
19		Program to find the factorial of a number	21	
20		Generate Fibonacci series of N terms	21	
21		Find the sum of all items in a list	22	
22		Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.	23	
23		Display the given pyramid with step number accepted from user.	24	
24		Count the number of characters (character frequency) in a string.	25	
25		Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.	26	
26		Accept a list of words and return length of longest word.	26	
27		Construct following pattern using nested loop.	27	

		<pre> *</pre>		
28		Generate all factors of a number.	28	
29		Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that finds area and perimeter of figures by different importing statements. (Include selective import of modules and import * statements)	29	
30		Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.	32	
31		Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.	33	
32		Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.	35	
33		Create a class Time with private attributes hour, minute and second. Overload '+' operator to find sum of 2 time	36	
34		Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no_of_pages. Write a program that	37	

		displays information about a Python book. Use base class constructor invocation and method overriding.		
35		Write a Python program to read a file line by line and store it into a list.	39	
36		Write a Python program to read each row from a given csv file and print a list of strings.	40	

## COURSE OUTCOME 1

### 1. Display future leap years from current year to a final year entered by

User.

#### SOURCE CODE

```

_startyear=int(input("Enter start year"))
lastyear=int(input("Enter last year"))
print("List of leap years")
for year in range(startyear,lastyear):
    if(year%4==0) and (year%100!=0) or (year%400==0):
        print(year)

```

#### OUTPUT

```

ccf@FISATPC0360: ~/abima
File Edit View Search Terminal Help
ccf@FISATPC0360:~$ cd abima
ccf@FISATPC0360:~/abima$ ls
abi.jpg      array7.c      pgm3.c
ABIMA.odt    biodata.html  positive.py
a.out        kozhikode.jpg 'Screenshot from 2021-11-05 09-41-31.png'
array1.c     largest.png   smallest.png
array2.c     leapyear.png  stack.c
array3.c     leapyear.py   'sum .png'
array4.c     'native place.html' TRTH.py
array5.c     pgm1.c
array6.c     pgm2.c
ccf@FISATPC0360:~/abima$ python3 leapyear.py
Enter start year2020
Enter last year2050
List of leap years
2020
2024
2028
2032
2036
2040
2044
2048
ccf@FISATPC0360:~/abima$

```

### 2.List comprehensions:

#### a.Generate positive list of numbers from a given list of integers.

##### SOURCE CODE

```

list1=[12,-1,-2,0,4,6,8]
for num in list1:
    if(num>=0):
        print(num)

```

## OUTPUT

```

ccf@FISATPC0360: ~/abima
File Edit View Search Terminal Help
ccf@FISATPC0360:~$ cd abima
ccf@FISATPC0360:~/abima$ ls
abi.jpg          kozhikode.jpg      'Screenshot from 2021-11-05 09-41-31.png'
ABIMA.odt        largest.png         smallest.png
a.out            leapyear.png        SQUAREOF.png
array1.c         'native place.html' squareof.py
array2.c         ORDINAL.png         square.png
array3.c         ordinal.py          square.py
array4.c         pgm1.c              stack.c
array5.c         pgm2.c              'sum .png'
array6.c         pgm3.c              TRTH.py
array7.c         positive.py
biodata.html     PYTHON
ccf@FISATPC0360:~/abima$ python3 positive.py
12
0
4
6
8
ccf@FISATPC0360:~/abima$

```

### b .Square of N numbers

#### SOURCE CODE

```

n=int(input("enter the range"))

for num in range(1,n+1):

    num=num*num

    print(num)

```

## OUTPUT

```

ccf@FISATPC0360: ~/abima
File Edit View Search Terminal Help
ccf@FISATPC0360:~$ cd abima
ccf@FISATPC0360:~/abima$ ls
abi.jpg          biodata.html        PYTHON
ABIMA.odt        kozhikode.jpg      'Screenshot from 2021-11-05 09-41-31.png'
a.out            largest.png         smallest.png
array1.c         leapyear.png        squareof.py
array2.c         'native place.html' square.png
array3.c         ORDINAL.png         square.py
array4.c         ordinal.py          stack.c
array5.c         pgm1.c              'sum .png'
array6.c         pgm2.c              TRTH.py
array7.c         pgm3.c
ccf@FISATPC0360:~/abima$ python3 squareof.py
enter the range6
1
4
9
16
25
36
ccf@FISATPC0360:~/abima$

```



**2c .Form a list of vowels selected from a given word.****SOURCE CODE**

```

s=input("Enter a string: ")

list=[]

for i in s:

    if i in "aeiouAEIOU":

        list.append(i)

print("vowels in the list are:")

print(list)

```

**OUTPUT**

```

ccf@FISATPC0360: ~/abima
File Edit View Search Terminal Help
ccf@FISATPC0360:~$ cd abima
ccf@FISATPC0360:~/abima$ ls
abi.jpg          largest.png      smallest.png
ABIMA.odt        leapyear.png    SQUAREOF.png
a.out            'native place.html'
array1.c          ORDINAL.png     squareof.py
array2.c          ordinal.py       square.png
array3.c          pgm1.c          square.py
array4.c          pgm2.c          stack.c
array5.c          pgm3.c          'sum .png'
array6.c          POSITIVE.png    TRTH.py
array7.c          positive.py      vowels1.py
biodata.html     PYTHON           vowels2.py
kozhikode.jpg    'Screenshot from 2021-11-05 09-41-31.png'
word.py
ccf@FISATPC0360:~/abima$ python3 vowels2.py
enter the statement alice
['a', 'i', 'e']
ccf@FISATPC0360:~/abima$

```

**d .List ordinal values of each element of a word.****SOURCE CODE**

```

list=['F','I','S','A','T']

for i in range(0,5):

    value=ord(list[i])

    print(value)

```

**OUTPUT**

```

ccf@FISATPC0360: ~/abima
File Edit View Search Terminal Help
ccf@FISATPC0360:~$ cd abima
ccf@FISATPC0360:~/abima$ ls
abi.jpg      array7.c      pgm3.c
ABIMA.odt    biodata.html  PYTHON
a.out        kozhikode.jpg 'Screenshot from 2021-11-05 09-41-31.png'
array1.c     largest.png   smallest.png
array2.c     leapyear.png  square.png
array3.c     'native place.html' square.py
array4.c     ordinal.py    stack.c
array5.c     pgm1.c        'sum .png'
array6.c     pgm2.c        TRTH.py
ccf@FISATPC0360:~/abima$ python3 ordinal.py
70
73
83
65
84
ccf@FISATPC0360:~/abima$

```

**3. Count the occurrences of each word in a line of text.****SOURCE CODE**

```

list1=[]

list2=[]

x=input("Enter a string:")

for i in x.split(" "):

    list1.append(i)

    if i not in list2:

        list2.append(i)

for i in list2:

    print(i,"\t",list1.count(i))

```

**OUTPUT**

```

stud@debian: ~/abima
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 occur.py
Enter a string:Timely completion of day to day tasks
Timely      1
completion  1
of           1
day         2
to          1
tasks       1
stud@debian:~/abima$

```

**4.Prompt the user for a list of integers. For all values greater than 100, store 'over' instead.**

### **SOURCE CODE**

```
list=[]

n=int(input("Enter the limit"))

print("Enter integer numbers")

for i in range(0,n):

    j=int(input())

    if(j>100):

        list.append("over")

    else:

        list.append(j)

print(list)
```

### **OUTPUT**

```
ccf@FISATPC0360: ~/abima
File Edit View Search Terminal Help
ccf@FISATPC0360:~$ cd abima
ccf@FISATPC0360:~/abima$ ls
abi.jpg          leapyear.png      SQUAREOF.png
ABIMA.odt        'native place.html' squareof.py
a.out            ORDINAL.png       square.png
array1.c         ordinal.py        square.py
array2.c         over.py           stack.c
array3.c         pgm1.c           'sum .png'
array4.c         pgm2.c           TRTH.py
array5.c         pgm3.c           vowels1.py
array6.c         POSITIVE.png     VOWELS2.png
array7.c         positive.py      vowels2.py
biodata.html     PYTHON           word.py
kozhikode.jpg   'Screenshot from 2021-11-05 09-41-31.png'
largest.png     smallest.png
ccf@FISATPC0360:~/abima$ python3 over.py
Enter the limit4
Enter integer numbers
56
110
23
200
[56, 'over', 23, 'over']
ccf@FISATPC0360:~/abima$
```

**5.Store a list of first names. Count the occurrences of 'a' within the list.**

### SOURCE CODE

```
list=["abi","adheena","lal"]

count=0

for word in list:

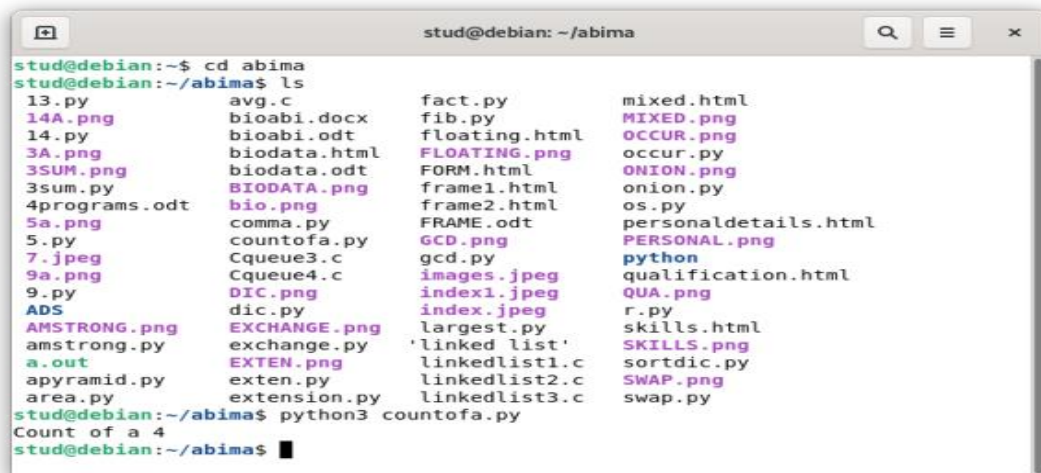
    for i in word:

        if i=='a':

            count+=1

print("Count of a")
```

### OUTPUT



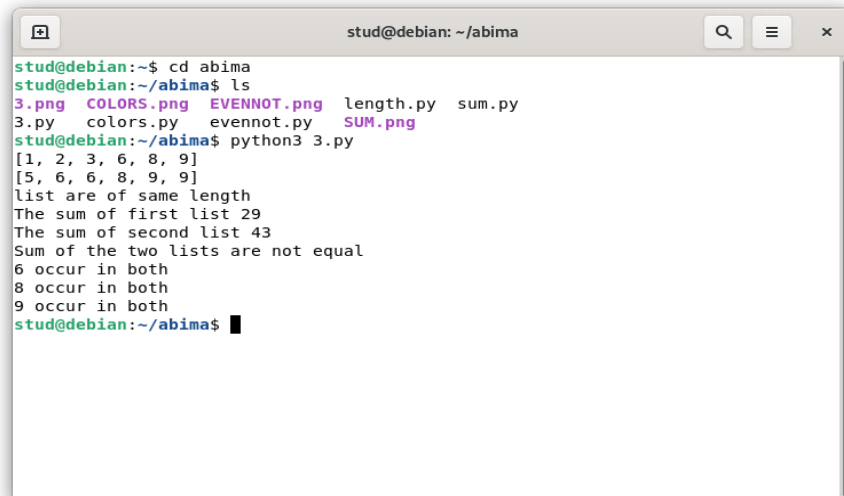
```
stud@debian: ~/abima
stud@debian:~/abima$ ls
13.py          avg.c          fact.py        mixed.html
14A.png        bioabi.docx    fib.py         MIXED.png
14.py          bioabi.odt     floating.html  OCCUR.png
3A.png         biodata.html  FLOATING.png  occur.py
3SUM.png       biodata.odt   FORM.html     ONION.png
3sum.py        BIODATA.png   frame1.html   onion.py
4programs.odt  bio.png       frame2.html   os.py
5a.png         comma.py      FRAME.odt     personaldetails.html
5.py           countofa.py   GCD.png       PERSONAL.png
7.jpeg         Cqueue3.c    gcd.py        python
9a.png         Cqueue4.c    images.jpeg   qualification.html
9.py           DIC.png       index1.jpeg   QUA.png
ADS            dic.py        index.jpeg    r.py
AMSTRONG.png  EXCHANGE.png  largest.py   skills.html
amstrong.py    exchange.py   'linked list' SKILLS.png
a.out          EXTEN.png     linkedlist1.c sortdic.py
apynamid.py    exten.py      linkedlist2.c SWAP.png
area.py        extension.py  linkedlist3.c swap.py
stud@debian:~/abima$ python3 countofa.py
Count of a 4
stud@debian:~/abima$
```

**6. Enter 2 lists of integers.Check**

- whether list are of same length
- whetherlist sums of same value
- whether any value occur in both.

**SOURCE CODE**

```
l1=[1,2,3,6,8,9]
l2=[5,6,6,8,9,9]
print(l1)
print(l2)
if (len(l1)==len(l2)):
    print("list are of same length")
else:
    print("list are of different length")
sum1=0
sum2=0
for i in range(len(l1)):
    sum1=sum1+l1[i]
print("The sum of first list",sum1)
for j in range(len(l2)):
    sum2=sum2+l2[j]
print("The sum of second list",sum2)
if(sum1==sum2):
    print("Sum of the two lists are equal")
else:
    print("Sum of the two lists are not equal")
flag=0
for i in l1:
    if i in l2:
        print(i,"occur in both")
        flag=1
if(flag==0):
    print("no common")
```

**OUTPUT**


```

stud@debian: ~$ cd abima
stud@debian:~/abima$ ls
3.png  COLORS.png  EVENNOT.png  length.py  sum.py
3.py   colors.py   evennot.py   SUM.png
stud@debian:~/abima$ python3 3.py
[1, 2, 3, 6, 8, 9]
[5, 6, 6, 8, 9, 9]
list are of same length
The sum of first list 29
The sum of second list 43
Sum of the two lists are not equal
6 occur in both
8 occur in both
9 occur in both
stud@debian:~/abima$

```

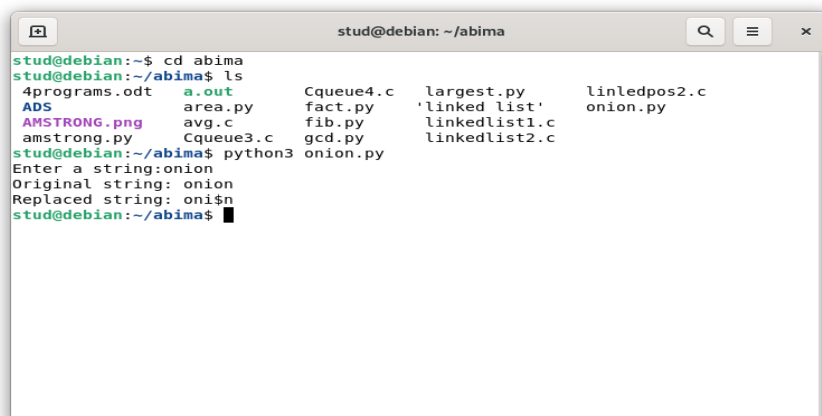
**7. Get a string from an input string where all occurrences of first character replaced with '\$', except first character. [eg: onion -> oni\$n]**

**SOURCE CODE**

```

str1=input("Enter a string:")
print("Original string:",str1)
char=str1[0]
str1=str1.replace(char,'$')
str1=char+str1[1:]
print("Replaced string:",str1)

```

**OUTPUT**


```

stud@debian: ~$ cd abima
stud@debian:~/abima$ ls
4programs.odt  a.out      Cqueue4.c  largest.py  linledpos2.c
ADS           area.py    fact.py    'linked list'  onion.py
AMSTRONG.png  avg.c      fib.py     linkedlist1.c
amstrong.py   Cqueue3.c  gcd.py     linkedlist2.c
stud@debian:~/abima$ python3 onion.py
Enter a string: onion
Original string: onion
Replaced string: oni$n
stud@debian:~/abima$

```

8. Create a string from given string where first and last characters exchanged.

[eg:python->nythop]

#### SOURCE CODE

```
s="python"
t=s[0]
t1=s[-1]
n=len(s)
s=t1+s[1:n-1]+t
print(s)
```

#### OUTPUT

```
stud@debian:~$ cd abima
stud@debian:~/abima$ ls
4programs.odt  area.py  fact.py  linkedlist1.c  os.py
ADS            avg.c    fib.py   linkedlist2.c
AMSTRONG.png  Cqueue3.c  gcd.py   linledpos2.c
amstrong.py   Cqueue4.c  largest.py  ONION.png
a.out         exchange.py 'linked list' onion.py
stud@debian:~/abima$ python3 exchange.py
nythop
stud@debian:~/abima$
```

9. Accept the radius from the user and find the area of the circle.

#### SOURCE CODE

```
x=input("Enter the radius")
x=int(x)
a=3.14*x*x
print(a)
```

#### OUTPUT

```
stud@debian:~$ cd abima
stud@debian:~/abima$ ls
area.py  fact.py  fib.py  kk  largest.py
stud@debian:~/abima$ python3 area.py
Enter the radius6
113.03999999999999
stud@debian:~/abima$
```

**10.Find the biggest of 3 numbers****SOURCE CODE**

```

a=input("Enter the number")
b=input("Enter the second number")
c=input("Enter the third number")
a=int(a)
b=int(b)
c=int(c)
if a>b:
    if a>c:
        print (a)
    else:
        print (c)
else:
    if b>c:
        print (b)
    else:
        print (c)

```

**OUTPUT**

```

stud@debian:~$ cd abima
stud@debian:~/abima$ ls
area.py  fact.py  fib.py  kk  largest.py
stud@debian:~/abima$ python3 largest.py
Enter the number5
Enter the second number7
Enter the third number10
10
stud@debian:~/abima$ █

```

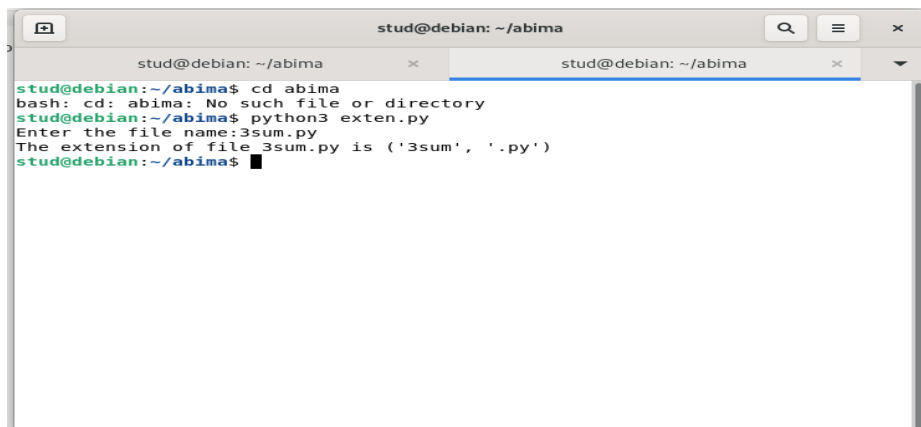
**11.Accept a file name from user and print extension of that.****SOURCE CODE**

```

import os
a=input("Enter the file name:")
print("The extension of file",a,"is",os.path.splitext(a))

```



**OUTPUT**


```

stud@debian: ~/abima
stud@debian: ~/abima
stud@debian:~/abima$ cd abima
bash: cd: abima: No such file or directory
stud@debian:~/abima$ python3 exten.py
Enter the file name:3sum.py
The extension of file 3sum.py is ('3sum', '.py')
stud@debian:~/abima$

```

**12.Create a list of colors from comma-separated color names entered by user.**

**Display first and last colors.**

**SOURCE CODE**

```

colors=[]
str=(input("Enter color names:"))
for i in str.split(','):
    colors.append(i)
print(colors)
print("first color:",colors[0],"Last color:",colors[-1])

```

**OUTPUT**

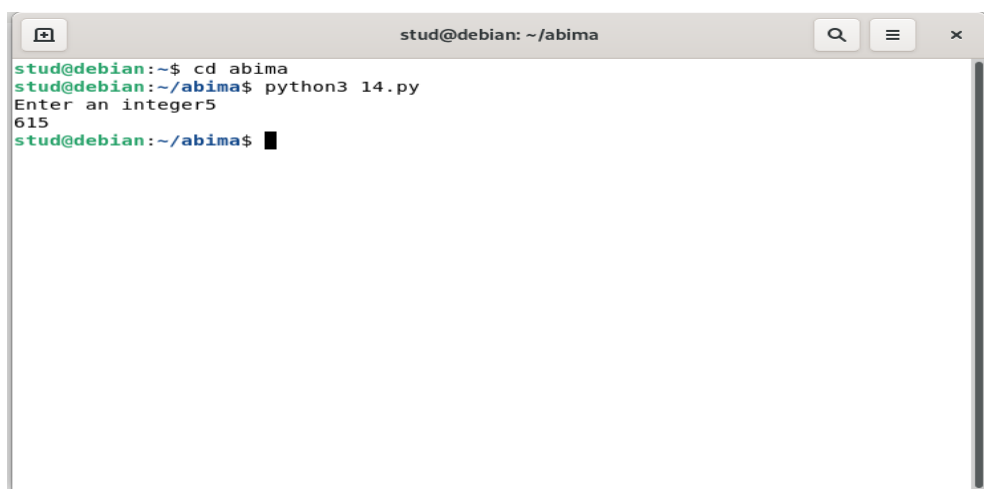

```

stud@debian:~$ cd abima
stud@debian:~/abima$ python3 13.py
Enter colors namewhite,blue,yellow
['white', 'blue', 'yellow']
First color: white Last color: yellow
stud@debian:~/abima$

```

**13. Accept an integer n and compute n+nn+nnn.****SOURCE CODE**

```
x=int(input("Enter an integer"))
n1=str(x)
n2=n1+n1
n3=n2+n1
result=int(n1)+int(n2)+int(n3)
print(result)
```

**OUTPUT**

```
stud@debian: ~/abima
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 14.py
Enter an integer5
615
stud@debian:~/abima$
```

**14. Print out all color from color-list1 not contained in color-list2****SOURCE CODE**

```
l1=["blue","orange","red","yellow"]
l2=["white","orange","green"]
l3=[]
for i in l1:
    if i not in l2:
        l3.append(i)
print(l3)
```

**OUTPUT**

```
stud@debian: ~/abima
stud@debian:~$ cd abima
stud@debian:~/abima$ ls
3.png  3.py  colors.py  even.py  length.py  SUM.png  sum.py
stud@debian:~/abima$ python3 colors.py
['blue', 'red', 'yellow']
stud@debian:~/abima$
```

**15. Create a single string separated with space from two strings by swapping the character at position 1.**

**SOURCE CODE**

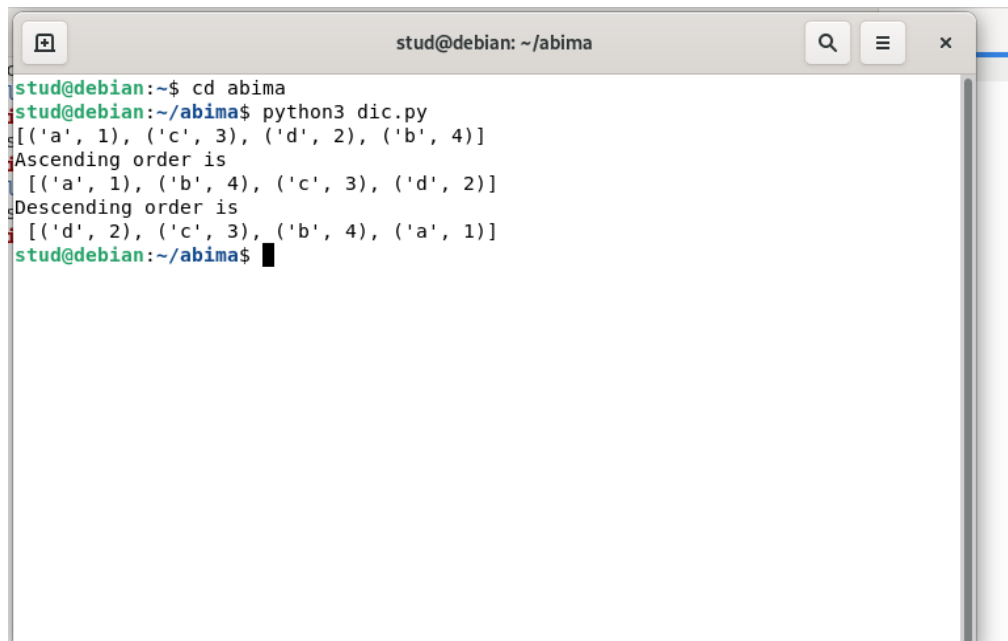
```
str1=input("Enter first string:")
str2=input("Enter second string:")
str3=str2[0]+str1[1:]+" "+str1[0]+str2[1:]
print(str3)
```

**OUTPUT**

```
stud@debian: ~/abima
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 swap.py
Enter first string:how are you
Enter second string:what about you
wow are you hhat about you
stud@debian:~/abima$
```

**16.Merge two dictionaries.****SOURCE CODE**

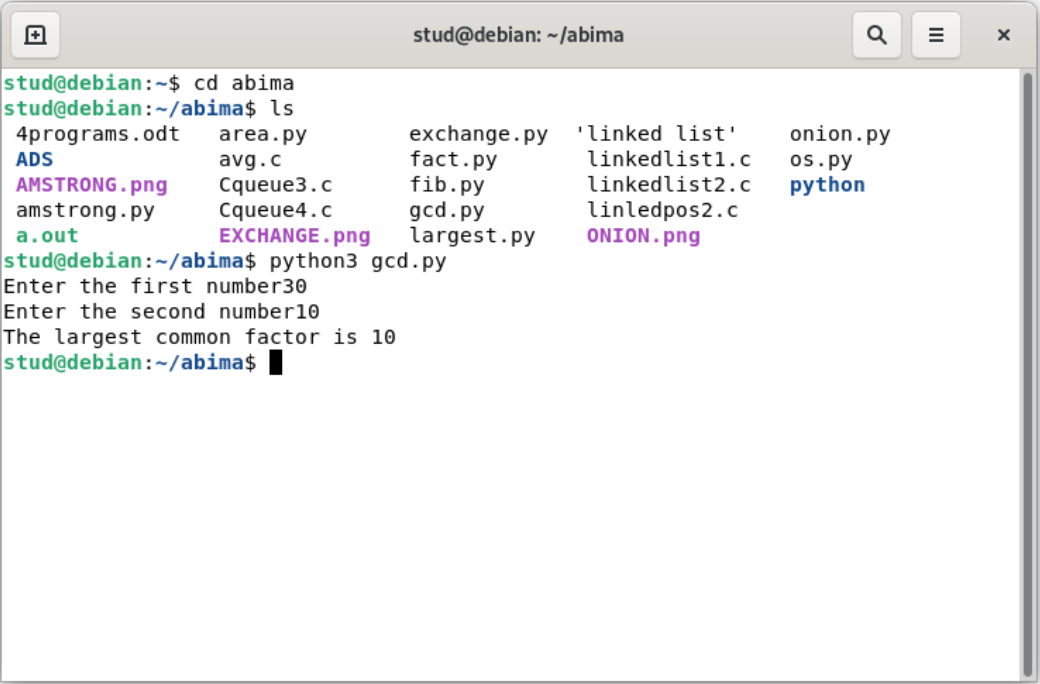
```
dict1={"a":1,"c":3,"d":2,"b":4}
l=list(dict1.items())
print(l)
l.sort()
print("Ascending order is\n",l)
l=list(dict1.items())
l.sort(reverse=True)
print("Descending order is\n",l)
```

**OUTPUT**A screenshot of a terminal window titled 'stud@debian: ~/abima'. The terminal shows the following commands and output:

```
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 dic.py
[('a', 1), ('c', 3), ('d', 2), ('b', 4)]
Ascending order is
[('a', 1), ('b', 4), ('c', 3), ('d', 2)]
Descending order is
[('d', 2), ('c', 3), ('b', 4), ('a', 1)]
stud@debian:~/abima$
```

**17.Find gcd of 2 numbers****SOURCE CODE**

```
a=int(input("Enter the first number"))
b=int(input("Enter the second number"))
if(a>b):
    x1=b
else:
    x1=a
for i in range(1,x1+1):
    if((a%i==0) and (b%i==0)):
        gcd=i
print("The largest common factor is",gcd)
```

**OUTPUT**

The screenshot shows a terminal window titled 'stud@debian: ~/abima'. The user enters the command 'cd abima' and then 'ls', which lists various files including '4programs.odt', 'ADS', 'AMSTRONG.png', 'a.out', 'amstrong.py', 'area.py', 'avg.c', 'Cqueue3.c', 'Cqueue4.c', 'EXCHANGE.png', 'exchange.py', 'fact.py', 'fib.py', 'gcd.py', 'largest.py', 'linked list', 'linkedlist1.c', 'linkedlist2.c', 'linledpos2.c', 'ONION.png', 'onion.py', 'os.py', and 'python'. The user then runs 'python3 gcd.py', which prompts for the first and second numbers (30 and 10) and outputs 'The largest common factor is 10'.

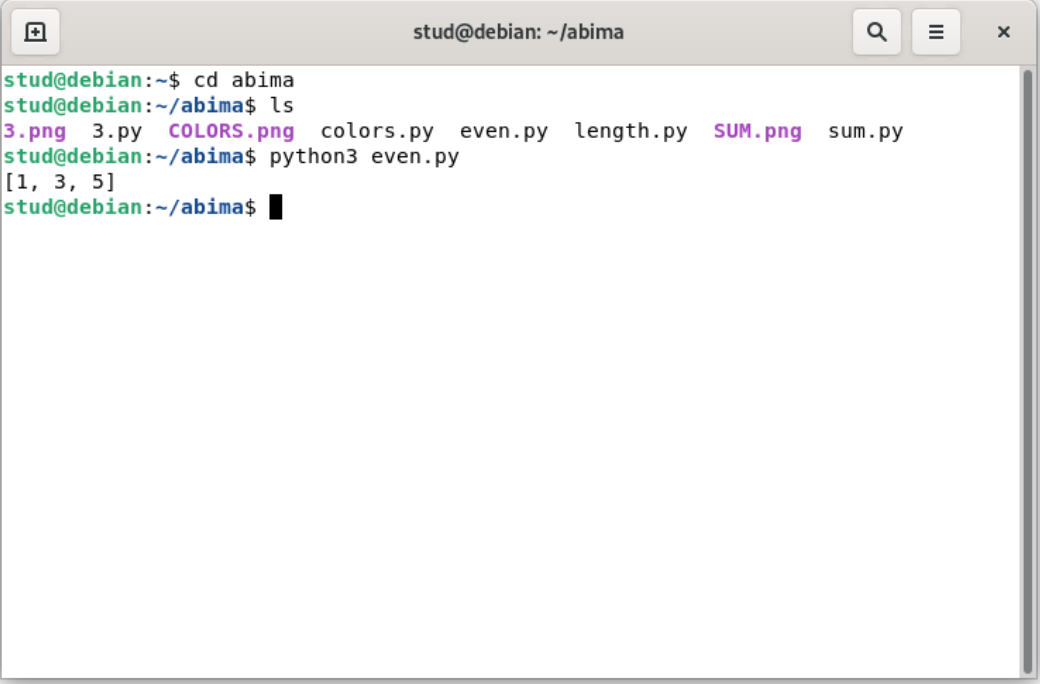
```
stud@debian:~$ cd abima
stud@debian:~/abima$ ls
4programs.odt  area.py      exchange.py  'linked list'  onion.py
ADS            avg.c        fact.py      linkedlist1.c  os.py
AMSTRONG.png   Cqueue3.c    fib.py       linkedlist2.c  python
amstrong.py    Cqueue4.c    gcd.py       linledpos2.c
a.out          EXCHANGE.png largest.py    ONION.png
stud@debian:~/abima$ python3 gcd.py
Enter the first number30
Enter the second number10
The largest common factor is 10
stud@debian:~/abima$
```

**18.From a list of integers,create a list removing even numbers.**

### SOURCE CODE

```
l1=[1,2,3,4,5,6]
l2=[]
for i in l1:
    if(i%2!=0):
        l2.append(i)
print(l2)
```

### OUTPUT

A terminal window titled 'stud@debian: ~/abima' showing the execution of a Python script. The user navigates to the 'abima' directory, lists files, and runs 'python3 even.py'. The output is '[1, 3, 5]'.

```
stud@debian:~$ cd abima
stud@debian:~/abima$ ls
3.png  3.py  COLORS.png  colors.py  even.py  length.py  SUM.png  sum.py
stud@debian:~/abima$ python3 even.py
[1, 3, 5]
stud@debian:~/abima$
```

**COURSE OUTCOME 2****19.Program to find the factorial of a number.****SOURCE CODE**

```
n=int(input('Enter the number:'))

fact=1

for i in range (1,n+1):

    fact=fact*i

print(fact)

stud@debian:~$ cd abima
stud@debian:~/abima$ ls
area.py fact.py fib.py kk largest.py
stud@debian:~/abima$ python3 fact.py
Enter the number7
5040
stud@debian:~/abima$ █
```

**20.Generate fibonacci series of N terms.****SOURCE CODE**

```
n=int(input('Enter a limit:'))

a=0

b=1

print(a)

print(b)

for i in range (2,n):

    c=a+b

    print(c)

    a=b

    b=c
```

**OUTPUT**

```

stud@debian:~$ cd abima
stud@debian:~/abima$ ls
area.py  fact.py  fib.py  kk  largest.py
stud@debian:~/abima$ python3 fib.py
Enter the number5
0
1
1
2
3
5
8
stud@debian:~/abima$ █

```

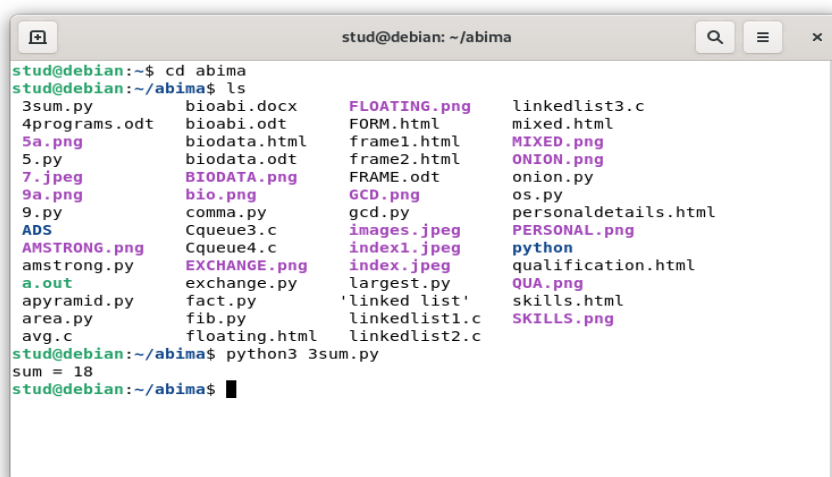
**21.Find the sum of all items in a list.**

**SOURCE CODE**

```

l1=[1,2,3,4,5,3]
sum=0
for i in range(len(l1)):
    sum=sum+l1[i]
print("sum =",sum)

```

**OUTPUT**


```

stud@debian:~$ cd abima
stud@debian:~/abima$ ls
3sum.py      bioabi.docx  FLOATING.png  linkedlist3.c
4programs.odt bioabi.odt   FORM.html     mixed.html
5a.png      biodata.html frame1.html    MIXED.png
5.py        biodata.odt  frame2.html    ONION.png
7.jpeg      BIODATA.png  FRAME.odt      onion.py
9a.png      bio.png      GCD.png        os.py
9.py        comma.py     gcd.py          personaldetails.html
ADS         Cqueue3.c   images.jpeg     PERSONAL.png
AMSTRONG.png Cqueue4.c   index1.jpeg     python
amstrong.py EXCHANGE.png index.jpeg       qualification.html
a.out       exchange.py  largest.py      QUA.png
apramid.py  fact.py      'linked list'   skills.html
area.py     fib.py       linkedlist1.c   SKILLS.png
avg.c       floating.html linkedlist2.c
stud@debian:~/abima$ python3 3sum.py
sum = 18
stud@debian:~/abima$ █

```



**22.Generate a list of four digit numbers in a given range with all their digits even and the number is a perfect square.**

### SOURCE CODE

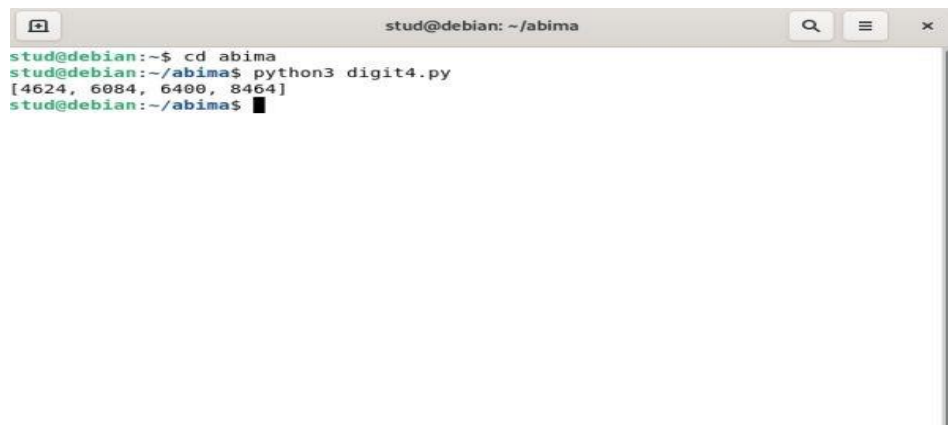
```
limit1=1000

limit2=9999

list1=[]
for i in range(limit1,limit2):
    j=i
    digit=[]
    while(i!=0):
        digit.append(i%10)
        i=int(i/10)
    count=0
    for n in digit:
        if n%2==0:
            count=count+1
    if count==4:
        for k in range(31,100):
            if((k**2)==j):
                list1.append(j)
                print(k)

print(list1)
```

### OUTPUT



```
stud@debian: ~/$ cd abima
stud@debian:~/abima$ python3 digit4.py
[4624, 6084, 6400, 8464]
stud@debian:~/abima$
```

**23.Display the given pyramid with step number accepted from user.**

### SOURCE CODE

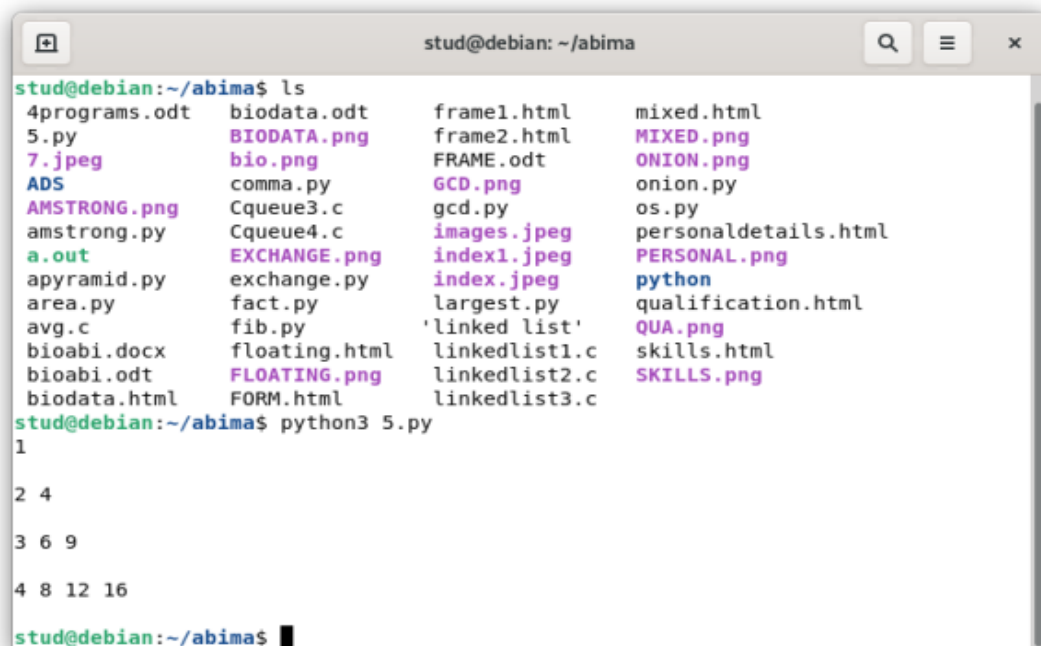
```
for j in range(1,5):

    for i in range(1,j+1):

        i=j*i
        print(i,end=" ")

    print("\n")
```

### OUTPUT



The screenshot shows a terminal window titled 'stud@debian: ~/abima'. The user runs 'ls' to list files, then 'python3 5.py' to execute the program. The output displays a pyramid pattern for step numbers 1 through 4.

```
stud@debian:~/abima$ ls
4programs.odt  biodata.odt  frame1.html  mixed.html
5.py          BIODATA.png  frame2.html  MIXED.png
7.jpeg        bio.png      FRAME.odt    ONION.png
ADS           comma.py     GCD.png      onion.py
AMSTRONG.png Cqueue3.c   gcd.py       os.py
amstrong.py  Cqueue4.c   images.jpeg  personaldetails.html
a.out        EXCHANGE.png index1.jpeg  PERSONAL.png
apyramid.py  exchange.py index.jpeg   python
area.py      fact.py     largest.py  qualification.html
avg.c        fib.py      'linked list' QUA.png
bioabi.docx  floating.html linkedlist1.c skills.html
bioabi.odt   FLOATING.png linkedlist2.c SKILLS.png
biodata.html FORM.html   linkedlist3.c

stud@debian:~/abima$ python3 5.py
1
2 4
3 6 9
4 8 12 16
stud@debian:~/abima$
```

**24.Count the number of characters (character frequency) in a string.****SOURCE CODE**

```
string=input("Enter a string:")
list1=[]
for i in string:
    if i not in list1:
        list1.append(i)
for i in list1:
    count=0
    for j in string:
        if(i==j):
            count=count+1
    print(i,"\t:",count)
```

**OUTPUT**

The screenshot shows a terminal window titled 'stud@debian: ~/abima'. The user enters the command 'cd abima' and then 'python3 6a.py'. The program prompts 'Enter a string:' and the user enters 'helloworld'. The output displays the character frequency for each character in the string: 'h' appears 1 time, 'e' appears 1 time, 'l' appears 3 times, 'o' appears 2 times, 'w' appears 1 time, 'r' appears 1 time, and 'd' appears 1 time. The terminal shows the command prompt 'stud@debian:~/abima\$' at the end.

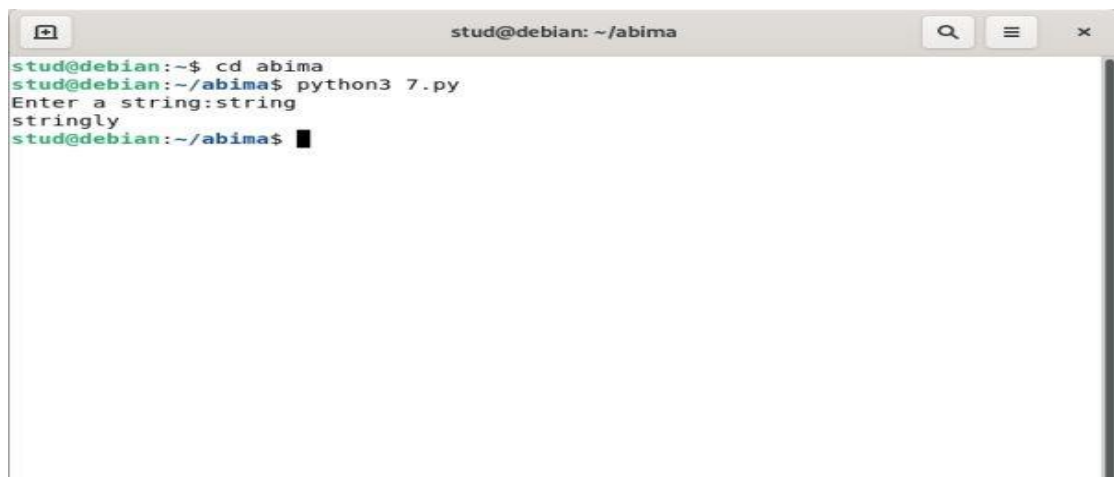
```
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 6a.py
Enter a string:helloworld
h      : 1
e      : 1
l      : 3
o      : 2
w      : 1
r      : 1
d      : 1
stud@debian:~/abima$
```

**25. Add 'ing' at the end of a given string. If it already ends with 'ing', then add 'ly'.**

#### SOURCE CODE

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

#### OUTPUT

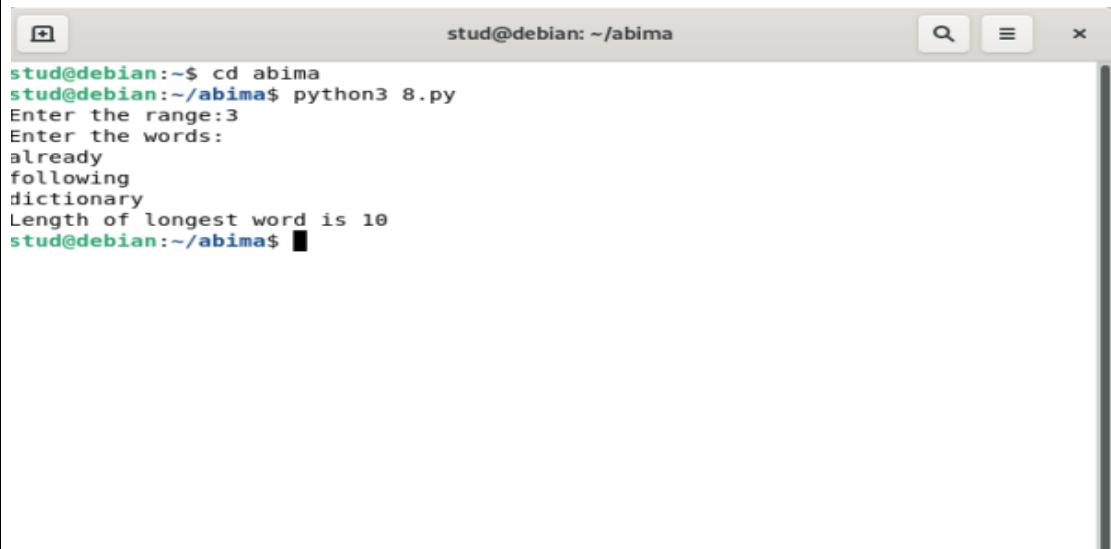


```
stud@debian: ~/abima
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 7.py
Enter a string:string
stringly
stud@debian:~/abima$
```

**26. Accept a list of words and return length of longest word.**

#### SOURCE CODE

```
lis=[]
n=int(input("Enter the range:"))
print("Enter the words:")
for i in range(0,n):
    lis.append(input(""))
longest=lis[0]
for i in range(1,n):
    if(len(lis[i])>len(longest)):
        longest=lis[i]
print("Length of longest word is",len(longest))
```

**OUTPUT**


```

stud@debian: ~/abima
stud@debian:~$ cd abima
stud@debian:~/abima$ python3 8.py
Enter the range:3
Enter the words:
already
following
dictionary
Length of longest word is 10
stud@debian:~/abima$

```

**27. Construct following pattern using nested loop.**

```

*
* *
* * *
* * * *
* * * * *
* * * * *
* * * *
* * *
* *
*

```

**SOURCE CODE**

```

for i in range(1,6):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")
for i in range(4,0,-1):
    for j in range(1,i+1):
        print("*",end=" ")
    print("\n")

```



**COURSE OUTCOME 3**

**29. Create a package graphics with modules rectangle, circle and sub-package 3D-graphics with modules cuboid and sphere. Include methods to find area and perimeter of respective figures in each module. Write programs that find area and perimeter of figures by different importing statements. (Include selective import of modules and import \* statements)**

**SOURCE CODE****Graphice\circle.py**

```
from math import pi
def area_circle(radius):
    return pi*radius*radius
def perimeter_circle(radius):
    return 2*pi*radius
```

**Graphics\rectangle.py**

```
def area_rec(length,width):
    return length*width
def perimeter_rec(length,width):
    return 2*(length+width)
```

**Graphics\tdgraphics\cuboid.py**

```
def area_cuboid(l,b,h):
    return 2*(l*h + b*h + l*b)
def volume_cuboid(l,b,h):
    return l*b*h
```

**Graphics\tdgraphics\sphere.py**

```
from math import pi
def area_sphere(radius):
    return 4*(pi*radius*radius)
def perimeter_sphere(radius):
    return 2*pi*radius
```

**graphics.py (driver code)**

```
import Graphics

from Graphics import circle,rectangle
from Graphics.tdgraphics import cuboid,sphere
from Graphics.circle import *

print("Area of a circle with radius 15 is : ",circle.area_circle(15))
print("Perimeter of a circle with radius 15 is ",circle.perimeter_circle(15))
print("\n")

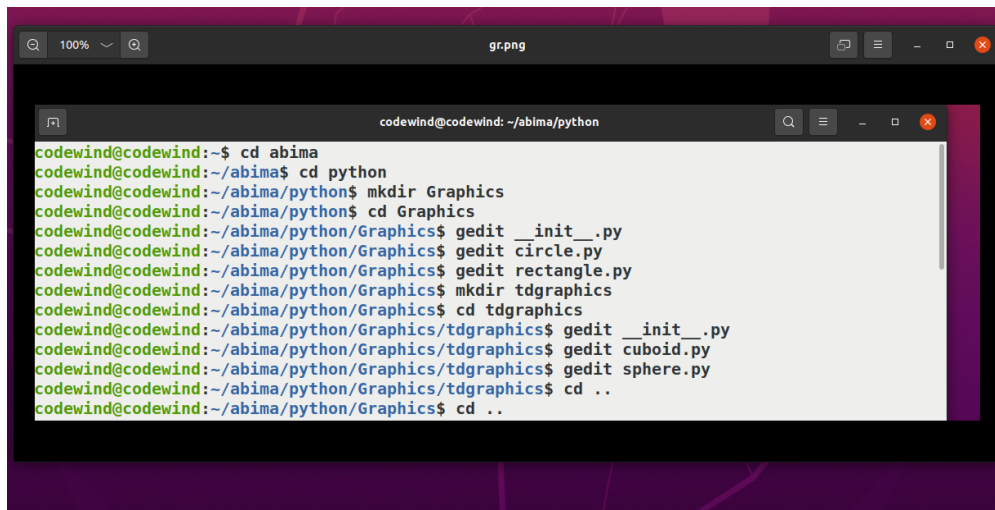
print("Area of a Rectangle with length and width 15 is :
      ",rectangle.area_rec(15,15))
print("Perimeter of a Rectangle with length and width 15 is :
      ",rectangle.perimeter_rec(15,15))
print("\n")

print("Area of a cuboid with length,width,height 15 is :
      ",cuboid.area_cuboid(15,15,15))
print("Volume of a cuboid with length,width,height 15 is :
      ",cuboid.volume_cuboid(15,15,15))
print("\n")

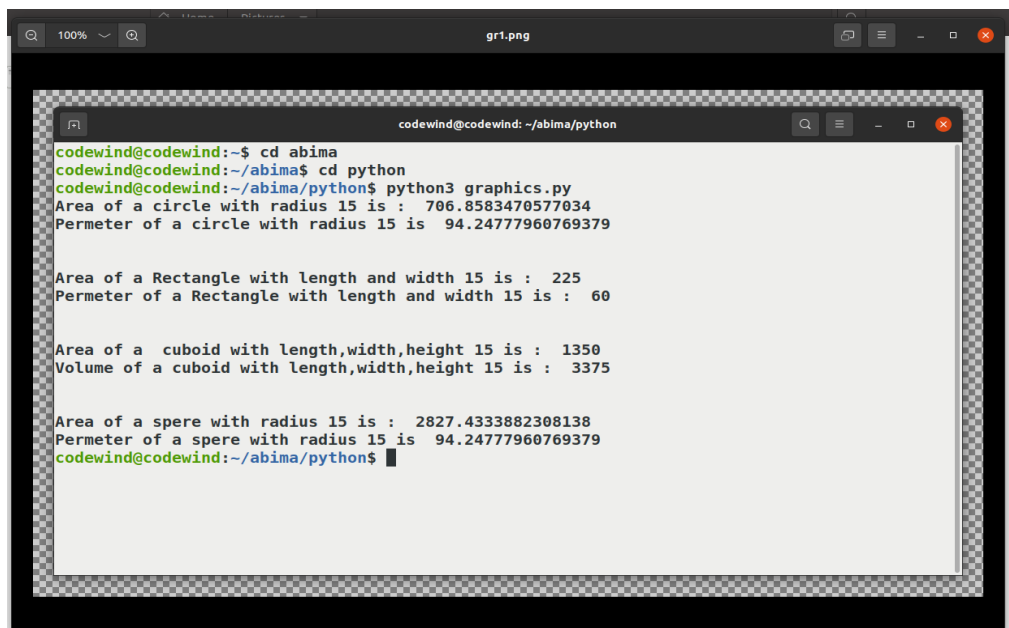
print("Area of a sphere with radius 15 is : ",sphere.area_sphere(15))
print("Perimeter of a sphere with radius 15 is ",sphere.perimeter_sphere(15))
```



## OUTPUT



```
codewind@codewind:~$ cd abima
codewind@codewind:~/abima$ cd python
codewind@codewind:~/abima/python$ mkdir Graphics
codewind@codewind:~/abima/python$ cd Graphics
codewind@codewind:~/abima/python/Graphics$ gedit __init__.py
codewind@codewind:~/abima/python/Graphics$ gedit circle.py
codewind@codewind:~/abima/python/Graphics$ gedit rectangle.py
codewind@codewind:~/abima/python/Graphics$ mkdir tdgraphics
codewind@codewind:~/abima/python/Graphics$ cd tdgraphics
codewind@codewind:~/abima/python/Graphics/tdgraphics$ gedit __init__.py
codewind@codewind:~/abima/python/Graphics/tdgraphics$ gedit cuboid.py
codewind@codewind:~/abima/python/Graphics/tdgraphics$ gedit sphere.py
codewind@codewind:~/abima/python/Graphics/tdgraphics$ cd ..
codewind@codewind:~/abima/python/Graphics$ cd ..
```



```
codewind@codewind:~$ cd abima
codewind@codewind:~/abima$ cd python
codewind@codewind:~/abima/python$ python3 graphics.py
Area of a circle with radius 15 is : 706.8583470577034
Perimeter of a circle with radius 15 is 94.24777960769379

Area of a Rectangle with length and width 15 is : 225
Perimeter of a Rectangle with length and width 15 is : 60

Area of a cuboid with length,width,height 15 is : 1350
Volume of a cuboid with length,width,height 15 is : 3375

Area of a sphere with radius 15 is : 2827.4333882308138
Perimeter of a sphere with radius 15 is 94.24777960769379
codewind@codewind:~/abima/python$
```

**COURSE OUTCOME 4**

**30.Create Rectangle class with attributes length and breadth and methods to find area and perimeter. Compare two Rectangle objects by their area.**

**SOURCE CODE**

```
class Rectangle:
    def __init__(self,l,b):
        self.l=l
        self.b=b
    def area(self):
        return self.l*self.b

    def perimeter(self):
        return 2*(self.l+self.b)

y1=Rectangle(15,5)
y2=Rectangle(25,5)
x=y1.area()
y=y2.area()
r=y1.perimeter()
v=y2.perimeter()
print("Area of first rectangle",x);
print("Area of second rectangle",y);
print("Perimeter of first rectangle",r);
print("Perimeter of second rectangle",v);
if(x>y):
    print("Area of first rectangle is greater")
else:
    print("Area of second rectangle is greater")
```

**OUTPUT**


```

codewind@codewind:~$ cd abima
codewind@codewind:~/abima$ python3 class1.py
Area of first rectangle 75
Area of second rectangle 125
Perimeter of first rectangle 40
Perimeter of second rectangle 60
Area of second rectangle is greater
codewind@codewind:~/abima$

```

**31.Create a Bank account with members account number, name, type of account and balance. Write constructor and methods to deposit at the bank and withdraw an amount from the bank.**

**SOURCE CODE**

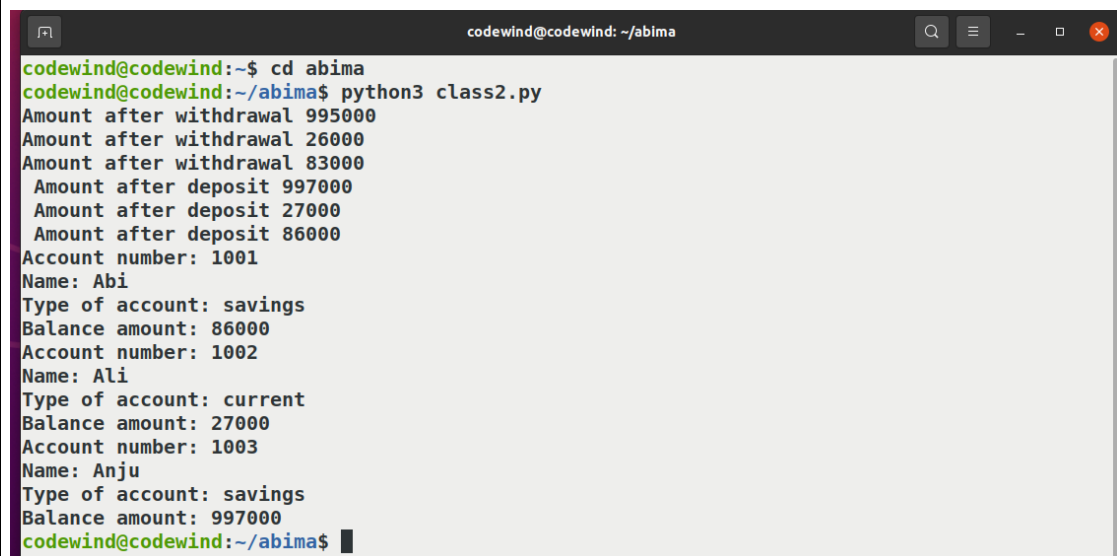
```

class Bank:
    def __init__(self,accountno,name,accounttype,bal):
        self.accountno=accountno
        self.name=name
        self.accounttype=accounttype
        self.bal=bal
    def withdraw(self,x):
        self.bal=self.bal-x
        print("Amount after withdrawal",self.bal)
    def deposit(self,y):
        self.bal=self.bal+y
        print(" Amount after deposit",self.bal)
    def display(self):
        print("Account number:",self.accountno)
        print("Name:",self.name)
        print("Type of account:",self.accounttype)
        print("Balance amount:",self.bal)
cus1=Bank(1001,"Abi","savings",90000)
cus2=Bank(1002,"Ali","current",30000)
cus3=Bank(1003,"Anju","savings",1000000)

```

```
cus3.withdraw(5000)
cus2.withdraw(4000)
cus1.withdraw(7000)
cus3.deposit(2000)
cus2.deposit(1000)
cus1.deposit(3000)
cus1.display()
cus2.display()
cus3.display()
```

## OUTPUT



```
codewind@codewind: ~$ cd abima
codewind@codewind: ~/abima$ python3 class2.py
Amount after withdrawal 995000
Amount after withdrawal 26000
Amount after withdrawal 83000
Amount after deposit 997000
Amount after deposit 27000
Amount after deposit 86000
Account number: 1001
Name: Abi
Type of account: savings
Balance amount: 86000
Account number: 1002
Name: Ali
Type of account: current
Balance amount: 27000
Account number: 1003
Name: Anju
Type of account: savings
Balance amount: 997000
codewind@codewind: ~/abima$
```

**32.Create a class Rectangle with private attributes length and width. Overload '<' operator to compare the area of 2 rectangles.**

**SOURCE CODE**

```
class Rectangle:

    def __init__(self,length,breadth):

        self.__length = length

        self.__breadth = breadth


    def __lt__ (self,rect2):

        if self.__length*self.__breadth < rect2.__length*rect2.__breadth:

            return True

        else:

            return False


l=int(input("Enter length of rectangle1: "))
b=int(input("Enter breadth of rectangle1: "))
rect1 = Rectangle(l,b)

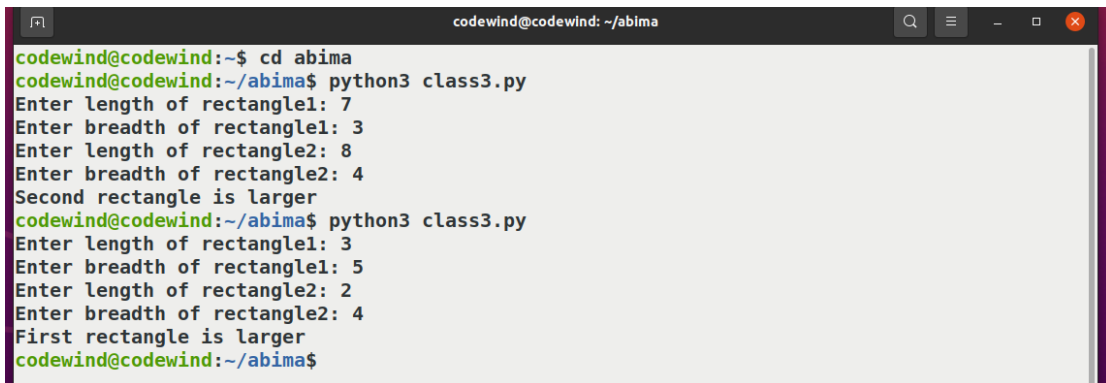

l=int(input("Enter length of rectangle2: "))
b=int(input("Enter breadth of rectangle2: "))
rect2 = Rectangle(l,b)


if rect1 < rect2:

    print("Second rectangle is larger")

else:

    print("First rectangle is larger")
```

**OUTPUT**


```

codewind@codewind: ~/$ cd abima
codewind@codewind:~/abima$ python3 class3.py
Enter length of rectangle1: 7
Enter breadth of rectangle1: 3
Enter length of rectangle2: 8
Enter breadth of rectangle2: 4
Second rectangle is larger
codewind@codewind:~/abima$ python3 class3.py
Enter length of rectangle1: 3
Enter breadth of rectangle1: 5
Enter length of rectangle2: 2
Enter breadth of rectangle2: 4
First rectangle is larger
codewind@codewind:~/abima$

```

**33. Create a class Time with private attributes hour, minute and second.**

**Overload '+' operator to find sum of 2 time.**

**SOURCE CODE**

```
class Time:
```

```
    def __init__(self,hr,min,sec):
```

```
        self.__hr=hr
```

```
        self.__min=min
```

```
        self.__sec=sec
```

```
    def __add__(self,t2):
```

```
        a=t1.__hr+t2.__hr
```

```
        b=t1.__min+t2.__min
```

```
        c=t1.__sec+t2.__sec
```

```
        print("Sum of 2 time is""-",a,":",b,":",c)
```

```
t1=Time(3,50,12)
```

```
t2=Time(4,33,10)
```

```
t1+t2
```

**OUTPUT**

```

codewind@codewind:~$ cd abima
codewind@codewind:~/abima$ python3 class4.py
Sum of 2 time is- 7 : 83 : 22
codewind@codewind:~/abima$ █

```

**34. Create a class Publisher (name). Derive class Book from Publisher with attributes title and author. Derive class Python from Book with attributes price and no\_of\_pages. Write a program that displays information about a Python book. Use base class constructor invocation and method overriding.**

**SOURCE CODE**

```

class Publisher(object):

    def __init__(self,name):

        self.name=name

    def display1(self):

        print(self.title)

        print(self.author)

class Book(Publisher):

    def __init__(self,name,title,author):

        super().__init__(name)

        self.title=title

        self.author=author

    def display2(self):

        #super().display1()

        print(self.title)

        print(self.author)

```

```
class Python(Book):  
  
    def __init__(self,name,title,author,price,no_of_pages):  
  
        super().__init__(name,title,author)  
  
        self.price=price  
  
        self.no_of_pages=no_of_pages  
  
    def display3(self):  
  
        super().display2()  
  
        print(self.price)  
  
        print(self.no_of_pages)  
  
d=Python("DC Publications","C programming","Anu Krishna",200,1000)  
  
d.display3()  
  
e=Python("Victoria Publications","Java programming","R Raj",500,2000)  
  
e.display3()  
  
f=Python("Navodhaya Publications","Mathematics","S.Santhosh",700,3000)  
  
f.display3()
```

## OUTPUT

```
codewind@codewind: ~/abima  
codewind@codewind:~$ cd abima  
codewind@codewind:~/abima$ python3 class5.py  
C programming  
Anu Krishna  
200  
1000  
Java programming  
R Raj  
500  
2000  
Mathematics  
S. Santhosh  
700  
3000  
codewind@codewind:~/abima$
```



**COURSE OUTCOME 5**

**35. Write a Python program to read a file line by line and store it into a list.**

**SOURCE CODE**

```
fp=open("text.txt",'r')

lines=[]

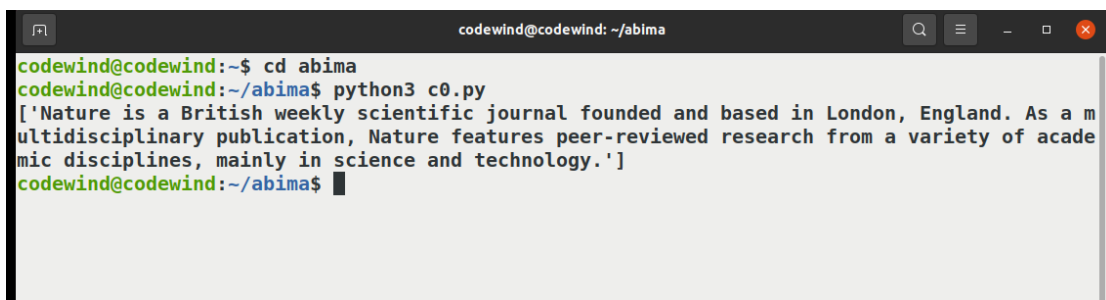
for line in fp:

    lines.append(line.strip())

print(lines)
```

**text.txt**

Nature is a British weekly scientific journal founded and based in London, England. As a multidisciplinary publication, Nature features peer-reviewed research from a variety of academic disciplines, mainly in science and technology.

**OUTPUT**A screenshot of a terminal window with a dark background. The title bar shows 'codewind@codewind: ~/abima'. The terminal content shows the following commands and output:

```
codewind@codewind:~$ cd abima
codewind@codewind:~/abima$ python3 c0.py
['Nature is a British weekly scientific journal founded and based in London, England. As a m
ultidisciplinary publication, Nature features peer-reviewed research from a variety of acade
mic disciplines, mainly in science and technology.']
codewind@codewind:~/abima$
```

**36. Write a Python program to read each row from a given csv file and print a list of strings.**

### **SOURCE CODE**

```
import csv

with open('m.csv', 'r') as file:
    reader = csv.reader(file)
    for row in reader:
        print(row)
```

### **m.csv**

Name, Subject, Marks

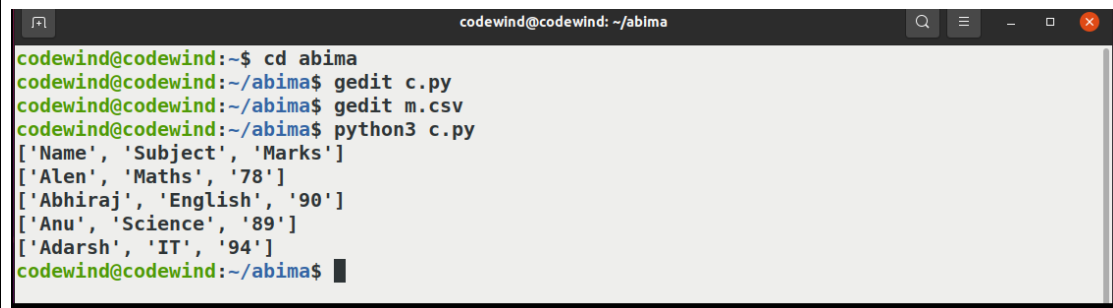
Alen, Maths, 78

Abhiraj, English, 90

Anu, Science, 89

Adarsh, IT, 94

### **OUTPUT**

A screenshot of a terminal window titled 'codewind@codewind: ~/abima'. The terminal shows the following commands and output:

```
codewind@codewind:~$ cd abima
codewind@codewind:~/abima$ gedit c.py
codewind@codewind:~/abima$ gedit m.csv
codewind@codewind:~/abima$ python3 c.py
['Name', 'Subject', 'Marks']
['Alen', 'Maths', '78']
['Abhiraj', 'English', '90']
['Anu', 'Science', '89']
['Adarsh', 'IT', '94']
codewind@codewind:~/abima$
```