

MALLA REDDY ENGINEERING COLLEGE

(Autonomous)

Maisammaguda, Dhulapally (post & via Kompally), Secunderabad-500 100.

Department of Information technology



Python Programming Lab

III Year B.Tech IT VI SEM (MR 17)

2019-2020

List of Programs:

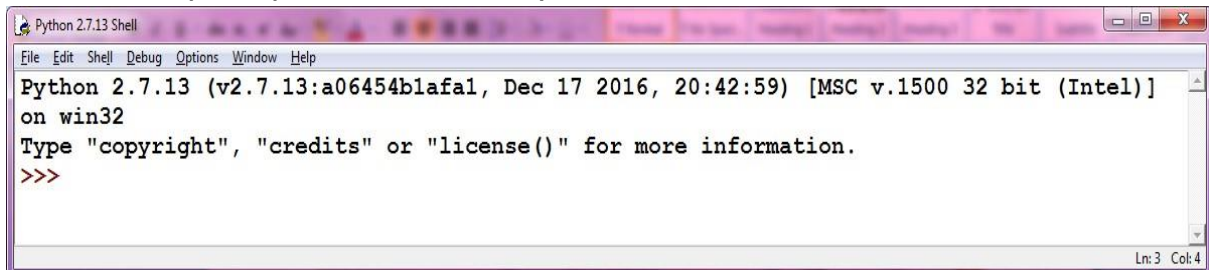
1. a) Run instructions in Interactive interpreter and a Python Script
b) Write a program to purposefully raise Indentation Error and correct it
2. a) Write a program to compute distance between two points taking input from the user
b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.
3. a) Write a Program for checking whether the given number is a even number or not.
b) Using a for loop, write a program that prints out the decimal equivalents of $1/2$, $1/3$,... , $1/10$.
c) Write a program using a for loop that loops over a sequence. What is sequence?
d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.
4. a) Find the sum of all the primes below two million. Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be: 1, 2,
3, 5, 8, 13, 21, 34, 55, 89, ...
b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms.
5. a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure
b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure.
6. a) Write a program combine lists that combines these lists into a dictionary.
b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?
7. a) Write a program to print each line of a file in reverse order.
b) Write a program to compute the number of characters, words and lines in a file.
8. a) Write a function ball_collide that takes two balls as parameters and computes if they are colliding.
Your function should return a Boolean representing whether or not the balls are colliding.
b) Find mean, median, mode for the given set of numbers in a list.
9. a) Write a function nearly_equal to test whether two strings are nearly equal.
b) Write a function dups to find all duplicates in the list.
c) Write a function unique to find all the unique elements of a list.
10. a) Write a function cumulative_product to compute cumulative product of a list of numbers.
b) Write a function reverse to reverse a list. Without using the reverse function.
11. Write a program a) defines a matrix and prints b) addition of two square matrices
c) multiplication of two square matrices.
12. a) Install packages requests, flask and explore them. using (pip).
b) Write a script that imports requests and fetch content from the page. Eg. (Wiki).
c) Write a simple script that serves a simple HTTPResponse and a simple HTML Page.
d) Write a program to implement class for ATM and include functions required for it.

Exercise 1 - Basics

a) Running instructions in Interactive interpreter and a Python Script.

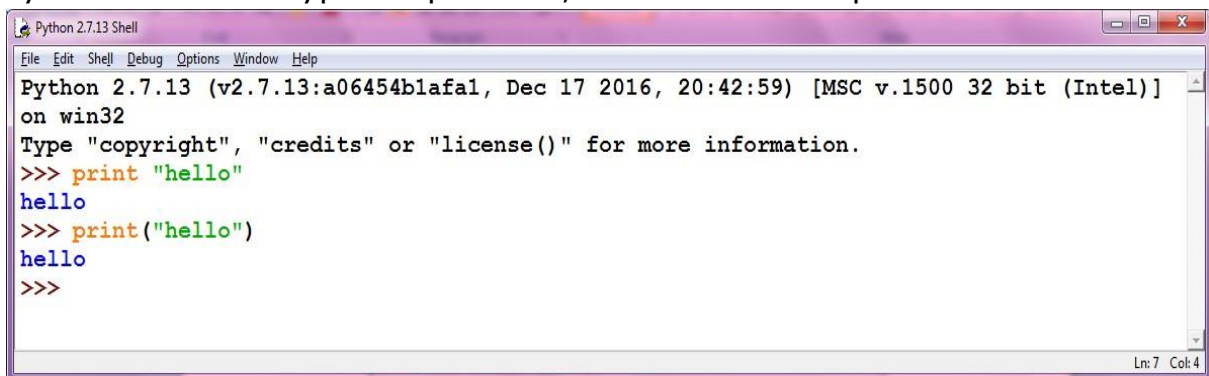
i. Running Python Interpreter:

Python comes with an interactive interpreter. When you type python in your shell or command prompt, the python interpreter becomes active with a >>> prompt and waits for your commands.



```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>>
```

Now you can type any valid python expression at the prompt. Python reads the typed expression, evaluates it and prints the result.

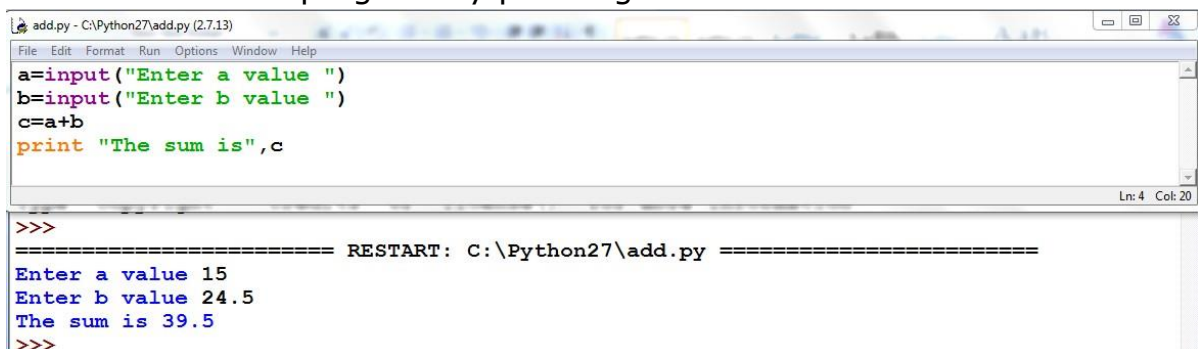


```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)]
on win32
Type "copyright", "credits" or "license()" for more information.
>>> print "hello"
hello
>>> print("hello")
hello
>>>
```

ii. Running Python Scripts in IDLE:

- Goto **File** menu click on New File (CTRL+N) and write the code and save add.py

```
a=input("Enter a value ")
b=input("Enter b value ")
c=a+b
print "The sum is",c
```
- And run the program by pressing F5 or Run→Run Module.



```
add.py - C:\Python27\add.py (2.7.13)
File Edit Format Run Options Window Help
a=input("Enter a value ")
b=input("Enter b value ")
c=a+b
print "The sum is",c
Ln: 4 Col: 20

>>>
===== RESTART: C:\Python27\add.py =====
Enter a value 15
Enter b value 24.5
The sum is 39.5
>>>
```

iii. Running python scripts in Command Prompt:

- Before going to run python27 folder in the command prompt.
- Open your text editor, type the following text and save it as hello.py.
print "hello"
- In the command prompt, and run this program by calling **python hello.py**. Make sure you change to the directory where you saved the file before doing it.

A screenshot of a Windows Command Prompt window. The title bar shows the path 'C:\Windows\system32\cmd.exe'. The command prompt shows the current directory as 'C:\Python27'. The user has entered the command 'python Hello.py', and the output is 'Hello, Mothilal'. The prompt is now 'C:\Python27>'.

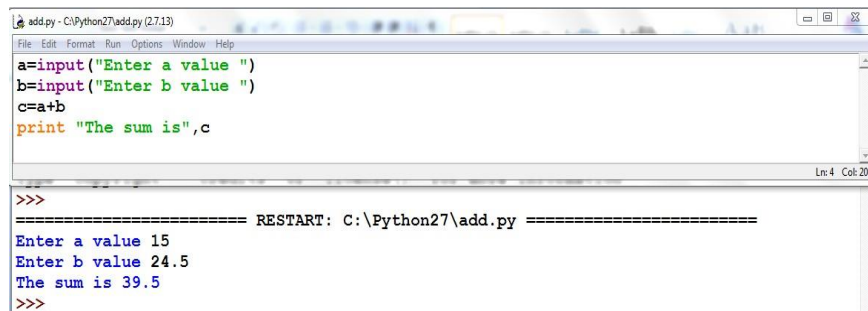
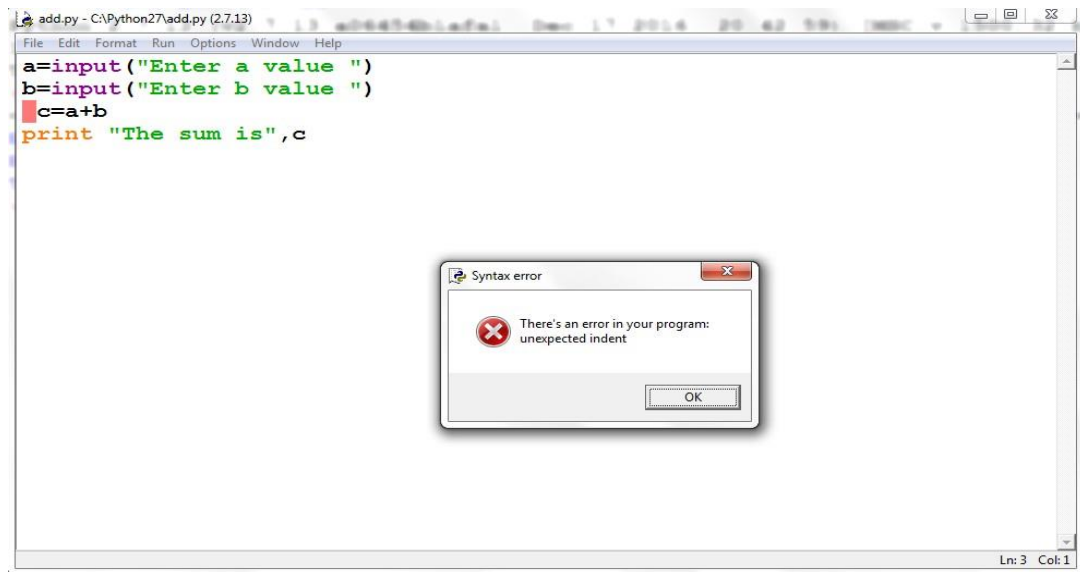
```
C:\Windows\system32\cmd.exe
C:\Python27>python Hello.py
Hello, Mothilal
C:\Python27>
```

b) Write a program to purposefully raise Indentation Error and correct it

Indentation

Code blocks are identified by indentation rather than using symbols like curly braces. Without extra symbols, programs are easier to read. Also, indentation clearly identifies which block of code a statement belongs to. Of course, code blocks can consist of single statements, too. When one is new to Python, indentation may come as a surprise. Humans generally prefer to avoid change, so perhaps after many years of coding with brace delimitation, the first impression of using pure indentation may not be completely positive. However, recall that two of Python's features are that it is simplistic in nature and easy to read.

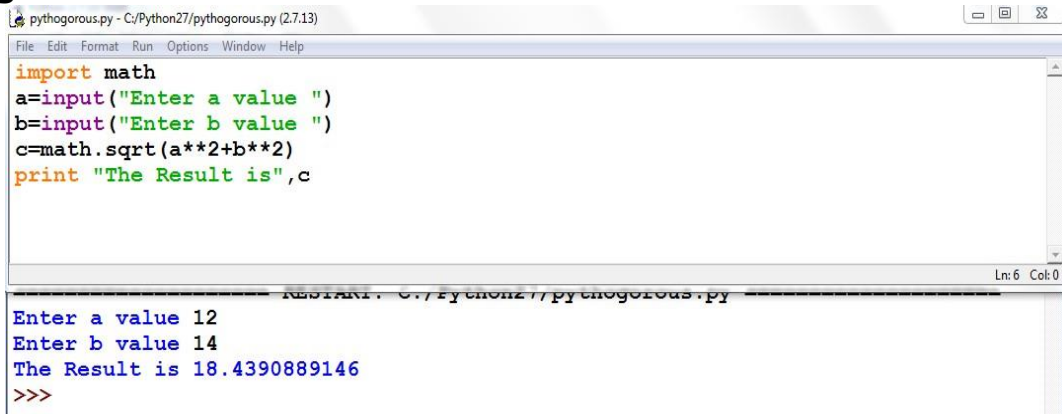
Python does not support braces to indicate blocks of code for class and function definitions or flow control. Blocks of code are denoted by line indentation. All the continuous lines indented with same number of spaces would form a block. Python strictly follow indentation rules to indicate the blocks.



Exercise 2 - Operations

a) Write a program to compute distance between two points taking input from the user (Pythagorean Theorem)

Program:



```
pythagorous.py - C:/Python27/pythagorous.py (2.7.13)
File Edit Format Run Options Window Help
import math
a=input("Enter a value ")
b=input("Enter b value ")
c=math.sqrt(a**2+b**2)
print "The Result is",c

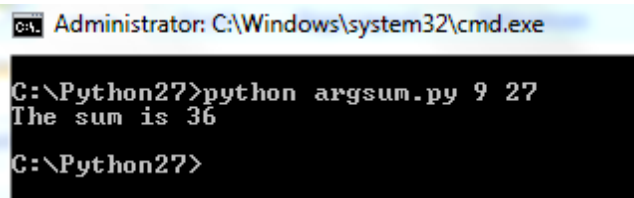
Enter a value 12
Enter b value 14
The Result is 18.4390889146
>>>
```

b) Write a program add.py that takes 2 numbers as command line arguments and prints its sum.

Program:

```
import sys
a=int(sys.argv[1])
b=int(sys.argv[2])
c=a+b
print "The sum is",c
```

Output:



```
C:\> Administrator: C:\Windows\system32\cmd.exe

C:\Python27>python argsum.py 9 27
The sum is 36

C:\Python27>
```

Exercise - 3 Control Flow

a) Write a Program for checking whether the given number is an even number or not.

Program:

```
a=input("Enter the number: ")
if a%2==0:
    print a,"is EVEN"
else:
    print a,"is NOT EVEN"
```

Output-1:

```
Enter the number: 15
15 is NOT EVEN
```

Output-2:

```
Enter the number: 24
24 is EVEN
```

b) Using a for loop, write a program that prints out the decimal equivalents of 1/2, 1/3, 1/4, . . . , 1/10

Program:

```
i=1
for i in range(1,11):
    print "Decimal equivalent value for 1/",i," is",1/float(i)
```

Output:

```
Decimal equivalent value for 1/1 is 1.0
Decimal equivalent value for 1/2 is 0.5
Decimal equivalent value for 1/3 is 0.333333333333
Decimal equivalent value for 1/4 is 0.25
Decimal equivalent value for 1/5 is 0.2
Decimal equivalent value for 1/6 is 0.166666666667
Decimal equivalent value for 1/7 is 0.142857142857
Decimal equivalent value for 1/8 is 0.125
Decimal equivalent value for 1/9 is 0.111111111111
Decimal equivalent value for 1/10 is 0.1
```

c) Write a program using a for loop that loops over a sequence.

What is sequence?

Sequence:

The Python for loop is an iterator based for loop. It steps through the items of lists, tuples, strings, the keys of dictionaries and other iterables. The Python for loop starts with the keyword "for" followed by an

arbitrary variable name, which will hold the values of the following sequence object.

```
for <variable> in <sequence>:  
    <statements>  
else:  
    <statements>
```

Program:

```
players=["kohli", "dhoni", "sachin", "sehwag", "Dravid"]  
for i in players:  
    print i
```

Output:

```
kohli  
dhoni  
sachin  
sehwag  
Dravid
```

d) Write a program using a while loop that asks the user for a number, and prints a countdown from that number to zero.

Program:

```
n=input("Enter the number for countdown: ")  
while 0<=n:  
    print n,  
    n=n-1
```

Output:

```
Enter the number for countdown: 15  
15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
```


Exercise - 4 - Control Flow – Continued

a) Find the sum of all the primes below two million.

Program:

Each new term in the Fibonacci sequence is generated by adding the previous two terms. By starting with 1 and 2, the first 10 terms will be:

1, 2, 3, 5, 8, 13, 21, 34, 55, 89, ...

```
n=input("Enter the range: ")
sum=0
for num in range(1,n+1):
    for i in range(2,num):
        if (num % i) == 0:
            break
    else:
        sum += num
print "Sum of prime numbers is", sum
```

Output:

```
Enter the range: 100
Sum of prime numbers is 1061
```

b) By considering the terms in the Fibonacci sequence whose values do not exceed four million, find the sum of the even-valued terms. **Program:**

```
n=input("Enter n value ")
f0=1
f1=2
sum=f1
print f0,f1,
for i in range(1,n-1):
    f2=f0+f1
    print f2,
    if f2%2==0:
        sum+=f2
    f0=f1
    f1=f2
print "\nThe sum of even Fibonacci numbers is", sum
```

Output:

```
Enter n value 10
1 2 3 5 8 13 21 34 55 89
The sum of even Fibonacci numbers is 44
```

Exercise - 5 - DS

a) Write a program to count the numbers of characters in the string and store them in a dictionary data structure

Program:

```
list=[]
for i in range(1,5):
    a=input("Enter the string ")
    list.append(a)
print "List is",list
dict={}
for i in list:
    dict[i]=len(i)
print "Dictionary is",dict
```

Output:

```
Enter the string "WELCOME"
Enter the string "TO"
Enter the string "PYTHON"
Enter the string "LAB"
List is ['WELCOME', 'TO', 'PYTHON', 'LAB']
Dictionary is {'PYTHON': 6, 'TO': 2, 'WELCOME': 7, 'LAB': 3}
```

b) Write a program to use split and join methods in the string and trace a birthday with a dictionary data structure

Program:

```
dob={"mothi":"12-11-1990","sudheer":"17-08-1991","vinay":"31-08-1988"}
str1=input("which person dob you want: ")
l=str1.split()
birth=""
for i in l:
    if i in dob.keys():
        name=i
print " ".join([name,"Birthday is",dob[name]])
```

Output:

```
which person dob you want: "i want mothi birthday"
mothi Birthday is 12-11-1990
```

Exercise - 6 DS - Continued

a) Write a program combine_lists that combines these lists into a dictionary.

Program:

```
subjects=['ENG','M1','M3','CP','PHY','CHE']
marks=[85,90,91,95,84,87]
z=zip(subjects, marks)
d=dict(z)
print d
```

Output:

```
{'CHE': 87, 'ENG': 85, 'PHY': 84, 'M1': 90, 'M3': 91, 'CP': 95}
```

b) Write a program to count frequency of characters in a given file. Can you use character frequency to tell whether the given file is a Python program file, C program file or a text file?

Program:

```
filename=input("Enter the filename: ")
f=open(filename, "r")
count=dict()
for line in f:
    for ch in line:
        if ch in count:
            count[ch]=count[ch]+1
        else:
            count[ch]=1
print count
f.close()
```

Output:

```
{' ': 36, '-': 2, ',': 3, '.': 3, 'E': 1, 'I': 1, 'P': 2, 'a': 18, 'c': 8, 'b': 3,
'e': 29, 'd': 8, 'g': 11, 'f': 2, 'i': 16, 'h': 12, 'k': 1, 'j': 1, 'l': 10, 'o':
11, 'n': 20, 'q': 1, 'p': 3, 's': 16, 'r': 13, 'u': 9, 't': 22, 'w': 3, 'v': 2,
'y': 6}
```

Exercise - 7 Files

a) Write a program to print each line of a file in reverse order.

Program:

```
filename=input("Enter the filename: ")
f=open(filename,"r")
for line in f:
    line2=""
    for ch in range(len(line)-1,-1,-1):
        line2=line2+line[ch]
    print line2
f.close()
```

Output:

Enter the filename: "welcome.txt"

.egaugnal gnitpircs detneiro-tcejbo dna evitcaretni ,deterpretni ,level-hgih
a si nohtyP
.elbadaer ylhgih eb ot dengised si nohtyP
.segaugnal rehto naht snoitcurtsnoc laticcatnys rewef sah ti dna
,noitautcnup esu segaungal rehto sa erehw yltneuqerf sdrowyek hsilgnE
sesu tI

b) Write a program to compute the number of characters, words and lines in a file.

Program:

```
filename=input("Enter the filename: ")
f=open(filename,"r")
l=w=c=0
for line in f:
    words=line.split()
    l=l+1
    for word in words:
        w=w+1
        for ch in word:
            c=c+1
print "No.of lines",l
print "No.of words",w
print "No.of characters",c
f.close()
```

Output:

Enter the filename: "welcome.txt"
No.of lines 3
No.of words 37
No.of characters 237

Exercise - 8 Functions

a) Write a function `ball_collide` that takes two balls as parameters and computes if they are colliding. Your function should return a Boolean representing whether or not the balls are colliding.

Hint: Represent a ball on a plane as a tuple of (x, y, r) , r being the radius. If (distance between two balls centers) \leq (sum of their radii) then (they are colliding)

Program:

```
import math
def ball_collide(x1,y1,r1,x2,y2,r2):
    status=False
    d=math.sqrt((x2-x1)**2-(y2-y1)**2)
    r=r1+r2
    if(d<r):
        status=True
    else:
        status=False
    return status
s=ball_collide(1,2,4,111,102,3)
print "Balls Collision is",s
s=ball_collide(1,2,2,101,102,3)
print "Balls Collision is",s
```

Output:

```
Balls Collision is False
Balls Collision is True
```

b) Find mean, median, mode for the given set of numbers in a list.

Program:

```
def mean(a):
    s=sum(a)
    m1=float(s)/len(a)
    print "Mean is",m1
def median(a):
    a.sort( )
    n=len(a)
    if n%2==0:
        m2=float((a[n/2]+a[(n-1)/2])/2)
    else:
        m2=b[(n-1)/2]
    print "Median is",m2
def mode(a):
    s1=set()
    uniq=[ ]
    for x in a:
        if x in s1:
            uniq.append(x)
        s1.add(x)
    print "Mode is",uniq
lst=[1,1,2,2,3,4,5,6]
mean(lst)
median(lst)
mode(lst)
```

Output:

```
Mean is 3.0
Median is 2.0
Mode is [1, 2]
```

Exercise - 9 Functions - Continued

a) Write a function `nearly_equal` to test whether two strings are nearly equal. Two strings `a` and `b` are nearly equal when `a` can be generated by a single mutation on `b`.

Program:

```
def mutate(word):
    out_list = []
    letters = 'abcdefghijklmnopqrstuvwxyz'
    #insert a character
    for i in range(len(word) + 1):
        for j in range(26):
            out_list.append(word[:i] + letters[j] + word[i:])
    #deleting a character
    for i in range(len(word)):
        out_list.append(word[:i] + word[i + 1:])
    #replace a character
    for i in range(len(word)):
        for j in range(26):
            out_list.append(word[:i] + letters[j] + word[i + 1:])
    #swapping a characters
    current_word = []
    out_word = ""
    for i in range(len(word) - 1):
        for j in range(i + 1, len(word)):
            #converting string into list
            cword = list(word)
            #Swapping of characters in a list
            cword[i], cword[j] = cword[j], cword[i]
            #converting list into string
            str1 = "".join(current_word)
            out_list.append(str1)
    return out_list

def nearly_equal(word1, word2):
    if len(word1) < len(word2):
        word1, word2 = word2, word1
    return word1 in mutate(word2)
else:
    return word1 in mutate(word2)

a = input("Enter First Word: ")
b = input("Enter Second Word: ")
print nearly_equal(a, b)
```

Output-1:

```
Enter First Word: "welcome"
Enter Second Word: "welcoe"
True
```

Output-2:

Enter First Word: "welcome"
Enter Second Word: "welcoome"
True

Output-3:

Enter First Word: "welcome"
Enter Second Word: "welcometoyou"
False

b) Write a function dups to find all duplicates in the list.**Program:**

```
def dup(a):  
    s=set()  
    d=set()  
    for i in a:  
        if i in s:  
            d.add(i)  
        else:  
            s.add(i)  
    print "Duplicate elements are",d  
dup([1,1,2,3,4,5,5,4,7,2])
```

Output:

Duplicate elements are set([1, 2, 4, 5])

c) Write a function unique to find all the unique elements of a list.**Program:**

```
def uni(a):  
    s=set()  
    d=set()  
    u=set()  
    for i in a:  
        if i in s:  
            d.add(i)  
        else:  
            s.add(i)  
    for i in a:  
        if i not in d:  
            u.add(i)  
    print "unique elements are",u  
uni([1,1,2,3,4,5,5,4,7,2])
```

Output:

unique elements are set([3, 7])

Exercise - 10 - Functions - Problem Solving

a) Write a function `cumulative_product` to compute cumulative product of a list of numbers.

Program:

```
def cumulative_product(a):  
    p=1  
    product=[]  
    for i in a:  
        p *=i  
        product.append(p)  
    print "cumulative_product is",product  
cumulative_product([1,2,3,4,5])
```

Output:

```
cumulative_product is [1, 2, 6, 24, 120]
```

b) Write a function `reverse` to reverse a list. Without using the `reverse` function.

```
def rev_list(a):  
    b=[]  
    for i in range(len(a)-1,-1,-1):  
        b.append(a[i])  
    print b  
rev_list([11,12,13,14,15])
```

Output:

```
[15, 14, 13, 12, 11]
```

Exercise 11 - Multi-D Lists

a) Write a program that defines a matrix and prints

```
r1=input("Enter A Matrix Rows:")
c1=input("Enter A Matrix Columns:")
a=[]
for i in range(r1):
    a.append([])
for i in range(r1):
    for j in range(c1):
        a[i].insert(j,input("Enter value:"))
print "a=",a
r2=input("Enter B Matrix Rows:")
c2=input("Enter B Matrix Columns:")
b=[]
for i in range(r2):
    b.append([])
for i in range(r2):
    for j in range(c2):
        b[i].insert(j,input("Enter value:"))
print "b=",b
```

Output:

```
Enter A Matrix Rows:3
Enter A Matrix Columns:3
Enter value:1
Enter value:2
Enter value:3
Enter value:4
Enter value:5
Enter value:6
Enter value:7
Enter value:8
Enter value:9
a= [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Enter B Matrix Rows:3
Enter B Matrix Columns:3
Enter value:1
Enter value:2
Enter value:3
Enter value:4
Enter value:5
Enter value:6
Enter value:7
Enter value:8
Enter value:9
b= [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

b) Write a program to perform addition of two square matrices

```
r1=input("Enter A Matrix Rows:")
c1=input("Enter A Matrix Columns:")
a=[]
for i in range(r1):
    a.append([])
for i in range(r1):
    for j in range(c1):
        a[i].insert(j,input("Enter value:"))
print "a=",a
r2=input("Enter B Matrix Rows:")
c2=input("Enter B Matrix Columns:")
b=[]
for i in range(r2):
    b.append([])
for i in range(r2):
    for j in range(c2):
        b[i].insert(j,input("Enter value:"))
print "b=",b
if r1==r2 and c1==c2:
    c=[]
    for i in range(r1):
        c.append([])
    for i in range(r1):
        for j in range(c1):
            c[i].insert(j,a[i][j]+b[i][j])
    for i in range(r1):
        for j in range(c1):
            print c[i][j],"\t",
        print ""
else:
    print "Addition is NOT Possible"
```

Output:

```
Enter A Matrix Rows:3
Enter A Matrix Columns:3
Enter value:1
Enter value:2
Enter value:3
Enter value:4
Enter value:5
Enter value:6
Enter value:7
Enter value:8
Enter value:9
a= [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

Enter B Matrix Rows:3
Enter B Matrix Columns:3
Enter value:4
Enter value:5
Enter value:6
Enter value:7
Enter value:8
Enter value:9
Enter value:1
Enter value:2
Enter value:3
b= [[4, 5, 6], [7, 8, 9], [1, 2, 3]]

5	7	9
11	13	15
8	10	12

c) Write a program to perform multiplication of two square matrices

Program:

```
r1=input("Enter A Matrix Rows:")
c1=input("Enter A Matrix Columns:")
a=[]
for i in range(r1):
    a.append([])
for i in range(r1):
    for j in range(c1):
        a[i].insert(j,input("Enter value:"))
print "a=",a
r2=input("Enter B Matrix Rows:")
c2=input("Enter B Matrix Columns:")
b=[]
for i in range(r2):
    b.append([])
for i in range(r2):
    for j in range(c2):
        b[i].insert(j,input("Enter value:"))
print "b",b
if c1==r2:
    c=[]
    for i in range(r1):
        c.append([])
    for i in range(r1):
        for j in range(c2):
            x=0
```

```

        for k in range(c1):
            x+=a[i][k]*b[k][j]
        c[i].insert(j,x)
    for i in range(r1):
        for j in range(c1):
            print c[i][j],"\t",
        print ""
else:
    print "Multiplication is NOT Possible"

```

Output:

```

Enter A Matrix Rows:3
Enter A Matrix Columns:3
Enter value:1
Enter value:2
Enter value:3
Enter value:4
Enter value:5
Enter value:6
Enter value:7
Enter value:8
Enter value:9
a= [[1, 2, 3], [4, 5, 6], [7, 8, 9]]
Enter B Matrix Rows:3
Enter B Matrix Columns:3
Enter value:4
Enter value:5
Enter value:6
Enter value:1
Enter value:2
Enter value:3
Enter value:7
Enter value:8
Enter value:9
b [[4, 5, 6], [1, 2, 3], [7, 8, 9]]
27  33  39
63  78  93
99  123 147

```

Exercise - 12 - Modules

a) Install packages requests, flask and explore them. Using (pip)

- The requests and flask Packages are downloaded from internet. To download install the packages follow the commands
- **Installation of requests Package:**
 - ∞ **Command:** cd C:\Python27\Scripts
 - ∞ **Command:** pip install requests
- **Installation of flask Package:**
 - ∞ **Command:** cd C:\Python27\Scripts
 - ∞ **Command:** pip install flask

b) Write a script that imports requests and fetch content from the page. Eg. (Wiki)

```
import requests
r = requests.get('https://www.google.com/')
print r.status_code
print r.headers['content-type']
```

Output:

```
200
text/html; charset=ISO-8859-1
```

c) Write a simple script that serves a simple HTTPResponse and a simple HTML Page

Program: 12c.py

```
from flask import Flask,render_template
app = Flask(__name__)
print(app)
@app.route("/")
def main():
    return render_template('w3.html')
if __name__ == "__main__":
    app.run(host="127.0.0.1",port=2500)
```

Program: w3.html

```
<!DOCTYPE html>
<html>
    <body>
        <a href="https://www.google.com">This is a link</a>
    </body>
</html>
```

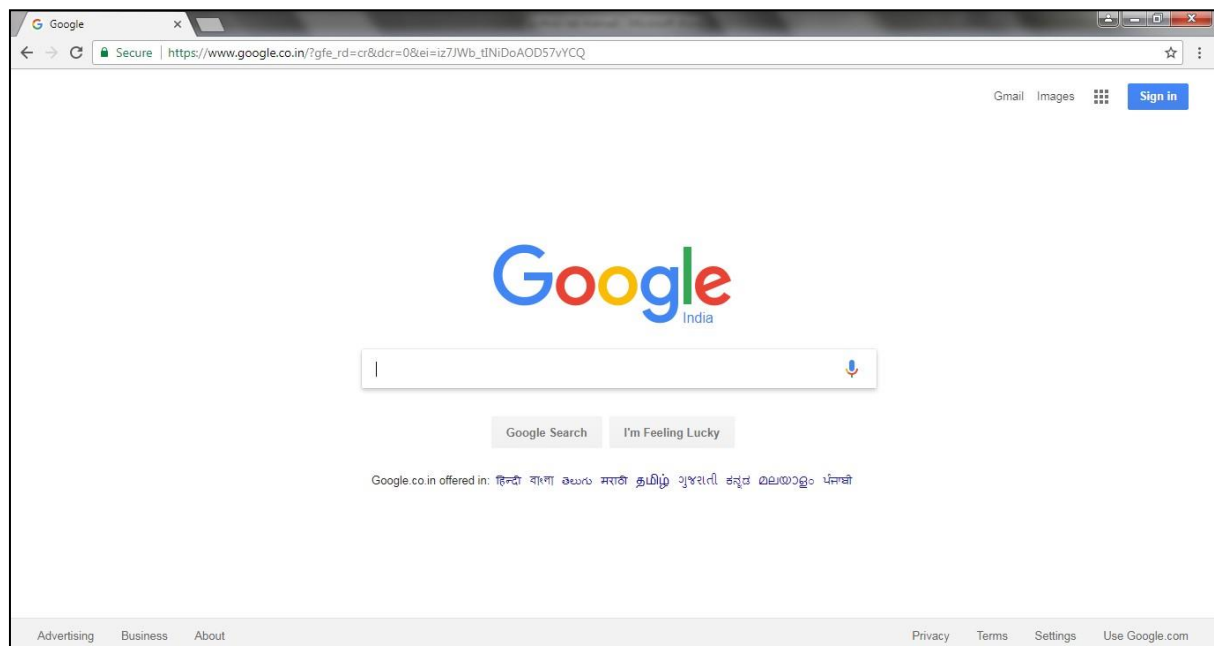
- Run the 12c.py python program

```
Python 2.7.13 Shell
File Edit Shell Debug Options Window Help
Python 2.7.13 (v2.7.13:a06454b1afa1, Dec 17 2016, 20:42:59) [MSC v.1500 32 bit (Intel)] on win32
Type "copyright", "credits" or "license()" for more information.
>>>
===== RESTART: C:\Python27\progs\12c.py =====
<Flask '12c'>
* Running on http://127.0.0.1:2500/ (Press CTRL+C to quit)
```

- Goto Browser <http://127.0.0.1:2500/>



- Click The Link in the shown and it will goto website link



d) Describe about Instance variable using ATM Machine Class

Program:

class ATM:

```
    def __init__(self):
        self.balance=0 #instance Variable print "New
        Account Created"
    def deposit(self):
        amount=input("Enter the amount to deposit: ")
        self.balance=self.balance+amount
        print "New Balance is",self.balance
    def withdraw(self):
        amount=input("Enter the amount to withdraw: ")
        if self.balance<amount:
            print "Insufficient Balance"
        else:
            self.balance=self.balance-amount
            print "New Balance is",self.balance
    def enquiry(self):
        print "The Balance is",self.balance
a=ATM()
a.deposit()
a.withdraw()
a.enquiry()
```

Output:

```
New Account Created
Enter the amount to deposit: 1200
New Balance is 1200
Enter the amount to withdraw: 950
New Balance is 250
Enter the amount to withdraw: 120
New Balance is 130
The Balance is 13
```