Installing Python, OpenCV and Anaconda in Ubuntu

Python:

1. Make sure we are in our home directory
   cd ~
2. Install Tools:
   sudo apt-get install build-essential cmake git pkg-config
3. Install File Format Libs
   sudo apt-get install libjpeg8-dev libtiff5-dev libjasper-dev libpng12-dev
4. Install Image -> GUI handler
   sudo apt-get install libgtk2.0-dev
5. Just in case we want to do any Video Processing:
   sudo apt-get install libavcodec-dev libavformat-dev libswscale-dev libv4l-dev
6. Install OpenCV Optimisation Libs
   sudo apt-get install libatlas-base-dev gfortran
7. Install Python 2.7
   sudo -H apt-get install python2.7-dev

Opencv

8. mkdir OpenCV-tmp
9. cd OpenCV-tmp
10. git clone https://github.com/Itseez/opencv.git
11. mv opencv opencv-3
12. mkdir build
13. cd build
14. cmake -D CMAKE_BUILD_TYPE=RELEASE –D CMAKE_INSTALL_PREFIX=/usr/local ../opencv-3
15. make
16. sudo make install

Anaconda

17. wget https://3230d63b5fc54e62148ec95ac804525aac4b6dba79b00b39d1d3.ssl.cf1.rackcdn.com/Anaconda-2.3.0-Linux-x86.sh
18. bash Anaconda-2.3.0-Linux-x86.sh
    Note: Type yes when prompted every time.
19. conda install opencv

To check installation

20. Python
21. import cv2
22. cv2.__version__
23. import numpy

24. import matplotlib
Note: There should not be any errors when import command is executed.

Installing Python, Opencv and Anaconda in Windows

1. Below Python packages are to be downloaded and installed to their default locations.

   1.1. Python-2.7.x. (http://python.org/ftp/python/2.7.5/python-2.7.5.msi)

   1.2. Numpy. (https://sourceforge.net/projects/numpy/files/NumPy/1.7.1/numpy-1.7.1-win32-superpack-python2.7.exe/download)

   1.3. Matplotlib (https://sourceforge.net/projects/matplotlib/files/matplotlib/matplotlib-1.3.0/matplotlib-1.3.0.win32-py2.7.exe/download?use_mirror=excellmedia)
   Install all packages into their default locations. Python will be installed to **C:/Python27/**.
2. After installation, open Python IDLE. Enter `import numpy` and make sure Numpy is working fine.
3. Download latest OpenCV release from https://sourceforge.net/projects/opencvlibrary/files/ and double-click to extract it.
4. Goto **opencv/build/python/2.7** folder.
5. Copy **cv2.pyd** to **C:/Python27/lib/site-packeges**.
6. Open Python IDLE and type following codes in Python terminal.
   >>> **import** cv2
   >>> **print** cv2.__version__

If the results are printed out without any errors, congratulations !!! You have installed OpenCV-Python successfully.

Anaconda for Windows (Optional)

7. Download Anaconda: https://www.continuum.io/downloads

8..Double click the Anaconda installer and follow the prompts to install to the default location.

NOTE: If you encounter any issues during installation, please temporarily disable your anti-virus software during install, then immediately re-enable it. If you have installed for all users, uninstall Anaconda and re-install it for your user only and try again.

Goto **opencv/build/python/2.7** folder. Copy **cv2.pyd** to C:\Program Files\Anaconda2\Lib\site-packages.

Preparing your Raspberry Pi for computer vision

1. Connect your Pi to an Internet modem or router with an Ethernet cable.
2. Run the following command to restart the networking service:
   sudo service networking restart
3. Make sure that Raspberry Pi is connected to the Internet by typing in the following command:

   ping –c4 www.google.com
4. Run the following commands in a sequence:

Advanced Package Tool (apt) is the utility that can be used to install and remove software in Debian and its variants. We need to use it to update the Pi software.

sudo apt-get update

This command synchronizes the package list from the source. Indexes of all the packages are refreshed. This command must be issued before we issue the upgrade command.

sudo apt-get upgrade

This will install the newest versions of the already installed software. Obsolete packages/utilities are not removed automatically. If the software is up to date, then it's left as it is.

sudo rpi-update

This command is used to upgrade the firmware. The kernel and firmware are installed as a Debian package, and hence, we will also get the updates. These packages are updated infrequently after extensive testing.

5.Now, we will need to install a few necessary packages and dependencies for OpenCV. Following is a list of packages we need to install. You just need to connect Pi to the Internet and type in sudo apt-get install <package-name>, where <package-name> is one of following packages:

libopencv-dev	libpng3	libdc1394-22-dev
build-essential	libpnglite-dev	libdc1394-22
libavformat-dev	zlib1g-dbg	libdc1394-utils
x264	zlib1g	libv4l-0
v4l-utils	zlib1g-dev	libv4l-dev
Ffmpeg	Pngtools	libpython2.6
libcv2.3	libtiff4-dev	python-dev
libcvaux2.3	libtiff4	python2.6-dev
libhighgui2.3	libtiffxx0c2	libgtk2.0-dev
python-opencv	libtiff-tools	libpngwriter0-dev
opencv-doc	libjpeg8	libpngwriter0c2
libcv-dev	libjpeg8-dev	libswscale-dev
libcvaux-dev	libjpeg8-dbg	libjpeg-dev

libhighgui-dev	libavcodec-dev	libwebp-dev
python-numpy	libavcodec53	libpng-dev
python-scipy	libavformat53	libtiff5-dev
python-matplotlib	libgstreamer0.10-0-dbg	libjasper-dev
python-pandas	libgstreamer0.10-0	libopenexr-dev
python-nose	libgstreamer0.10-dev	libgdal-dev
v4l-utils	libxine1-ffmpeg	python-tk
libgtkglext1-dev	libxine-dev	python3-dev
libpng12-0	libxine1-bin	python3-tk
libpng12-dev	libunicap2	python3-numpy
libpng++-dev	libunicap2-dev	libeigen3-dev

For example, if you want to install x264, you have to type sudo apt-get install x264. This will install the necessary package. Similarly, you can install all of the aforementioned packages in like manner. If a package is already installed on Pi, it will show the following message:

```
pi@pi02 ~ $ sudo apt-get install x264
Reading package lists... Done
Building dependency tree
Reading state information... Done
x264 is already the newest version.
0 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
```

In such cases, don't worry. The package you wanted to install has already been installed, and it is up to date. Just proceed with the installation of all the other packages in the list one-by-one.

6. Finally, install OpenCV for Python by using the following command:

sudo apt-get install python-opencv

This is the easiest way to install OpenCV for Python. However, there is a problem with this. Raspbian repositories may not always contain the latest version of OpenCV.

Basic Python Programs

1. Write a Python program to swap two numbers.

```
a=input('Enter value of A :')
b=input('Enter value of B :')
temp=a
a=b
```

```
b=temp
print "After Swapping"
print "A :",a
print "B :",b
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac1_1.py
Enter value of A : 4
Enter value of B : 2
After Swapping
A : 2
B : 4
```

2. Write a Python program to find the greatest of 3 numbers

```
a=input('Enter first number :')
b=input('Enter second number :')
c=input('Enter third number :')
if (a>b) and (a>c) :
        largest=a
elif (b>a) and (b>c) :
        largest=b
else :
        largest=c
print "The Largest number is :",largest
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac2_1.py
Enter first number :2
Enter second number :5
Enter third number :4
The Largest number is :5
```

3. Write a Python program to convert days entered into months and days

```
days = input('Enter Days :')
months=days/30
days=days%30
print 'Months :',months
print 'Days :',days
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac3_1.py
Enter Days :65
Months: 2
Days: 5
```

4. Write a Python program to find the roots of a Quadratic equation

```
import cmath
# To take coefficient input from the users
a=input('Enter first number :')
b=input('Enter second number :')
c=input('Enter third number :')
```

```
# calculate the discriminant
d = (b**2) - (4*a*c)
# find two solutions
root1 = (-b-cmath.sqrt(d))/(2*a)
root2 = (-b+cmath.sqrt(d))/(2*a)
print 'The solutions are'
print 'Root1 :',root1
print 'Root2 :',root2
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac4_1.py
Enter first number:1
Enter first number:3
Enter first number:2
'The solutions are
Root1 :-1.0
Root2 :-2.0
```

5. Write a Python program to evaluate the power function using while loop.

```
base=input("Enter the base number: ")
max=input("Enter the maximum exponent : ")
counter = 1
while counter<=max:
      res = base**counter
      counter = counter+1
print “Ans:”,res
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac5_1.py
Enter the base number: 2
Enter the maximum exponent : 5
 Ans: 32
```

6. Write a Python program to evaluate the factorial function using while loop.

```
num=input('Enter num :')
fact=1
if (num<0) :
        print 'sorry, Number is negative'
elif (num==0) :
        print 'Factorial of 0 is 1'
else :
        for i in range(1,num+1) :
                fact=fact*i
        print 'Factorial of ',num,' is ',fact
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac6_1.py
Enter num :5
Factorial of  5 is 120
```

7. Write a Python program to evaluate the sum of integers between a and b entered by the user that is divisible by 3.

```
a =input('Enter a :')
b =input('Enter b :')
sum=0
for i in range(a,b) :
        if (i%3==0) :
                sum=sum+i
print 'sum of all integers between ',a,' and ',b,' that is divisible by 3.'
print 'sum :',sum
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac7_1.py
Enter a:5
Enter b:10
sum of all integers between  5  and 10 that is divisible by 3.
sum:15
```

Practical Set : 2

1. Write a Python program to print the list of odd numbers until the number entered by the user

```
num = input('Enter number :')
print 'list of odd numbers until ',num,'is :'
for i in range (1,num+1):
        if(i%2==1):
                print i
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac1_2.py
Enter number :10
list of odd numbers until  10 is :
1
3
5
7
9
```

2. Write a Python program to print the series of prime numbers until the number entered by the user.

```
num = input('Enter number :')
print 'Series of all Prime Numbers until ',num,'is :'
for i in range (2,num+1):
        status=True
        for j in range (2,i):
                if(i%j==0):
                        status=False
                        break
        if(status!=False):
                print i
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac2_2.py
Enter number :10
```

Series of all Prime Numbers until 5 is :
2
3
5

3. Insert a string and the output should be the sum of the numbers separated by special character

```
str=raw_input("Enter the string :")
sum=0
for i in str.split('#'):
        sum=sum+int(i)
print ‘sum: ’,sum
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac3_2.py
Enter the string :1#2.3#2.4#1
sum : 6.7
```

4(a).Write a PYTHON program to get the following patterns:

1
2 3
4 5 6

```
k=1
for i in range(1,4):
        for j in range(1,i+1):
                print k,
                k=k+1
        print ' '
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac4a_2.py
        1
        2 3
        4 5 6
```

4(b). Write a PYTHON program to get the following patterns:

#

! !

# # #

```
for i in range(1,4):
        for k in range(1,4-i):
                print ' ',
        for j in range(1,i+1):
                if(i%2==1):
                        print '#',
                else:
                        print '!',
        print ' '
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac4b_2.py
          #
        ! !
       # # #
```

5. Write a Python program to create a very simple pizza-ordering menu. At this pizzeria, there's only one kind of pizza you can order: cheese pizza with no toppings. Your choices are what size of pizza, and how many of them. You'll need to figure out the total price of the order. A small pizza costs Rs. 200, a medium Rs. 350, and a large Rs. 500.

```
print 'Let me help you order a pizza'
print 'What size would you like ?'
print ' A : small(200)'
print ' B : medium(350)'
print ' C : large(500)'
print ' '
choice=raw_input('Please enter A or B or C :')
if(choice=='A' or choice=='a'):
        print 'OK, small , that is 200 Rs. each'
        cost=200
if(choice=='B' or choice=='b'):
        print 'OK, medium , that is 350 Rs. each'
        cost=350
if(choice=='C' or choice=='c'):
        print 'OK, large , that is 500 Rs. each'
        cost=500
print ' '
quantity=input('How many would you like ?')
total_cost=cost*quantity
print ' Your total cost is Rs.',total_cost
print ' '
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac5_2.py
Let me help you order a pizza
What size would you like ?
A : small(200)
B : medium(350)
C : large(500)
Please enter A or B or C : A
OK, small , that is 200 Rs. Each
How many would you like ?: 3
Your total cost is Rs. 600
```

6. Write a program called time.py that prompts the user to enter the current time in HH:MM format, and then prints a message that states the time in a sentence. For example, a sample run of your program might look something like this:

```
time=raw_input('What is the time please (HH:MM)? :')
list=time.split(':')
print ' '
print "Thanks! it is now ",list[1],"minutes after ",list[0],"o'clock"
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac6_2.py
What is the time please (HH:MM)? : 06:15
Thanks! it is now 15 minutes after  6 o'clock"
What is the time please (HH:MM)? : 6:40
Thanks! it is now 40 minutes after  6 o'clock"
```

Practical Set : 3

1. Write a Python function to sum all the numbers in a list.

```
def add(a,b):
        return a+b
List=[]
length=input("Enter the size of List :")
for i in range(1,length+1):
        element=input("Enter element :")
        List.append(element)
print List
ans=reduce(add,List)
print "sum of all the numbers in a list is :",ans
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac1_3.py
Enter the size of List :5
Enter element :5
Enter element :5
Enter element :2
Enter element :3
Enter element :6
[5, 5, 2, 3, 6]
sum of all the numbers in a list is :21
```

2. Write a Python function to multiply all the numbers in a list.

```
def multiplication(a,b):
        return a*b
List=[]
length=input("Enter the size of List :")
for i in range(1,length+1):
        element=input("Enter element :")
        List.append(element)
print List
ans=reduce(multiplication,List)
```

```
print "multiplication of all the numbers in a list is :",ans
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac2_3.py
Enter the size of List :4
Enter element :10
Enter element :5
Enter element :2
Enter element :3
[10, 5, 2, 3]
sum all the numbers in a list is :300
```

3. Write a Python function to calculate the factorial of a number (a non-negative integer). The function accepts the number as an argument.

```
def factorial(n):
        fact=1
        for i in range(1,n+1):
                fact=fact*i
        return fact
num=input("Enter number :")
ans=factorial(num)
print "Factorial of ",num,"is :",ans
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac3_3.py
Enter number : 3
Factorial of 3 is : 6
```

4. Write a Python function that takes a number as a parameter and check the number is prime or not.

```
def prime(num):
        if(num>1):
                for i in range(2,num):
                        if not (num%i):
                                return False
        else:
                return False
        return True
number=input("Enter the number :")
ans=prime(number)
if(ans==True):
        print "The number is prime"
else:
        print "The number is not prime"
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac4_3.py
Enter the number :3
The number is prime
[comp@localhost]$ python Prac4_3.py
Enter the number :9
```

```
The number is not prime"'
```

5. Write a Python program to print the even numbers from a given list.

```
def even_no(x):
        return x%2==0
list=[]
n=input("Enter the size of List :")
for i in range(1,n+1):
        element=input("Enter Element :")
        list.append(element)
print list
even_number=filter(even_no,list)
print "All Even numbers are:",even_number
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac5_3.py
Enter the size of List :1
Enter Element :1
Enter Element :4
Enter Element :2
Enter Element :3
Enter Element :9
[1, 4, 2, 3, 9]
All Even numbers are: [2, 4]
```

6. Write a Python function to create and print a list where the values are square of numbers between 1 and 30 (both included).

```
fun=lambda x:x*x
list=[]
for i in range(1,31):
        list.append(fun(i))
print list
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac6_3.py
[1, 4, 9, 16, 25, 36, 49, 64, 81, 100, 121, 144, 169, 196, 225, 256, 289, 324, 361, 400, 441, 484,
529, 576, 625, 676, 729, 784, 841, 900]
```

7. Write a Python function that that prints out the first n rows of Pascal's triangle.

```
def pascal_triangle(x):
        if(x==0):
                return [1]
        else:
                List=pascal_triangle(x-1)
                ans=[1]
                for i in range(len(List)-1):
                        ans.append(List[i]+List[i+1])
                ans.append(1)
                return ans
n=input("Enter number of Rows :")
for i in range(0,n):
```

```
        print pascal_triangle(i)
```

*************OUTPUT**************

```
 [comp@localhost]$ python Prac7_3.py
[1]
[1,1]
[1,2,1]
[1,3,3,1]
[1,4,6,4,1]
```

Practical Set : 4

1-a. Take two numbers from the users and write it to the file.

```
import sys
fopen=open("test.txt","w")
fopen.write("First Number  :"+str(sys.argv[1]))
fopen.write("\nSecond Number :"+str(sys.argv[2]))
fopen.close()
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac1_4.py 2 3
```

text.txt

```
First Number  :1
Second Number :4
```

1-b. Read the number from the file

```
import sys
fopen=open("test.txt","r")
line = fopen.readline()
#print line
while line:
        print line
        line = fopen.readline()
fopen.close()
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac1b_4.py
First Number  :1
Second Number  :4
```

2. Write a Python program to Check the file is empty or not.

```
import os
import sys
try:
   s = os.stat('test.txt')
   if s.st_size == 0:
      print "The file {} is empty".format('test.txt')
      sys.exit(1)
except OSError as e:
   print e
   sys.exit(2)
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac2_4.py
The file test.txt is empty
```

3. Write a Python program to convert its contents in a upper case and in lower case. Also print it on screen

```
import sys
fopen=open('test.txt','r')
content=fopen.read()
upper=open('Upper.txt','w')
upper.write(content.upper())
print content.upper()

lower=open('Lower.txt','w')
lower.write(content.lower())
print content.lower()
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac3_4.py
```

Upper.txt

```
HELLO
GOOD MORNING
HOW ARE YOU?
```

Lower.txt

```
hello
good morning
how are you?
```

4. Write a Python program to count the number of white spaces in a given text file.

```
file=open('test.txt','r')
content=file.read()
count=content.split()
print 'White spaces =',(len(count))
```

**************OUTPUT**************

```
 [comp@localhost ]$ python Prac4_4.py
White spaces = 6
```

5. Write a Python program to replace a specific word in file with a new word by creating a new file.

```
file=open('test.txt','r')
content=file.read()
file.close()
text1=raw_input("Enter text you want to search :")
text2=raw_input("Enter text you want to replace :")
file=open('test.txt','w')
content=content.replace(text1,text2)
file.write(content)
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac5_4.py
Enter text you want to search : India
```

Enter text you want to replace : Bharat

test.txt

Jai Bharat

6. Write a Python program to concate two text files.

```
inputFileNames=['file1.txt','file2.txt']
with open('concate.txt','w') as outputFile:
        for i in inputFileNames:
                with open(i) as inputFile:
                        for line in inputFile:
                                outputFile.write(line)
```

**************OUTPUT**************

```
 [comp@localhost ]$ python Prac6_4.py
file1.txt: Jai Hind
file2.txt: Jai Bharat
```

concate.txt

Jai Hind
Jai Bharat

7. Write a Python program to reverse the contents of the file and print it on the terminal.

```
import os
file1=open("test.txt",'r')
file2=open("reverse.txt",'w')
content=file1.read()
file2.write(content[::-1])
print content[::-1]
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac7_4.py
```

reverse.txt

dnih iaj

8. Write a Python program to count total number of lines in a given file.

```
file=open('test.txt','r')
content=file.read()
line=content.splitlines()
print 'Number of Lines :',(len(line))
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac8_4.py
Number of Lines : 3
```

9. Write a Python program to count the occurrence of all the characters present in the given text file.

```
from string import ascii_lowercase
with open('test.txt') as file:
        content = file.read().strip()
        dictionary = {}
        for x in ascii_lowercase:
                dictionary[x] = content.count(x)
        print dictionary
```

**************OUTPUT**************

test.txt

```
jai hind
[comp@localhost]$ python Prac9_4.py
{'a': 1, 'c': 0, 'b': 0, 'e': 0, 'd': 1, 'g': 1, 'f': 0, 'i': 2, 'h': 1, 'k': 0, 'j': 0, 'm': 0, 'l': 2, 'o': 0, 'n': 1, 'q': 0,
'p': 0, 's': 0, 'r': 0, 'u': 0, 't': 0, 'w': 0, 'v': 0, 'y': 0, 'x': 0, 'z': 0}
'''
```

10. Write a Python program to take file name as an input from the user and print the following things related to it:
a. Print the status of existence of the file i.e. yes if file presents and no otherwise
b. Print the type of the file i.e. Directory file, or regular file
c. Print the size of the file

```
import os
file_name=raw_input("Enter File Name :")
print "Status of File exisitance is :",os.path.exists(file_name)
print "Type of the file is :"
if(os.path.isfile(file_name)):
        print "regular file"
elif(os.path.isdir(file_name)):
        print "Directory file"
print "Size of file is :",os.path.getsize(file_name)
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac10_4.py
Enter File Name : test.txt
Status of File exisitance is : True
Type of the file is :
regular file
Size of file is : 31
```

11. Write a Python program to evaluate the sum of integers stored in a text file and prints the result on the terminal.

```
file=open("Int.txt",'r')
sum=0
for line in file:
        sum=sum+int(line)
print "Sum of all integers:",sum
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac11_4.py
Sum of all integers: 23
```

12. Write a Python program to read specific line from the given non empty text file.

```
file = open('test.txt','r')
x=int(input("Enter Line Number :"))
line = file.read().splitlines()[x]
print line
```

**************OUTPUT**************

test.txt

```
Jai hind
Jai Bharat
```

```
[comp@localhost]$ python Prac12_4.py
Enter Line Number :1
Jai Bharat
```

13. Write a Python program to print the names of all the files in the current working directory.

```
import os
files = filter(os.path.isfile, os.listdir( os.curdir ) )
print files
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac13_4.py
['Prac1a_4.py', 'Prac1b_4.py', 'Prac2_4.py', 'Prac3_4.py', 'test.txt', 'Upper.txt', 'Lower.txt',
'Prac4_4.py', 'Prac5_4.py', 'Prac6_4.py', 'file1.txt', 'file2.txt', 'concate.txt', 'Prac7_4.py',
'reverse.txt', 'Prac8_4.py', 'Prac9_4.py', 'Prac10_4.py', 'Prac11_4.py', 'Int.txt', 'Prac12_4.py',
'Prac13_4.py']
```

14. Write a Python program to take path from the user and do the followings:
a. Remove all the files inside that directory
b. Create a subdirectory inside it
c. Remove the newly created empty directory in it

```
import os
path=raw_input("Enter path: ")
os.chdir(path)
'''for f in os.listdir('.'):
        os.remove(f)
        print f'''
print "creating new directory"
os.mkdir("new")
files=[f for f in os.listdir('.')]
print files
print "after removing directory"
os.rmdir("new")
files=[f for f in os.listdir('.')]
print files
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac14_4.py
Enter path: /home/comp/abc
menu.html
text.txt
menu.txt
Enter path: /home/comp/abc
creating new directory
['menu.html', 'text.txt', 'menu .txt', 'new']
after removing directory
['menu.html', 'text.txt', 'menu .txt']
```

Practical Set : 5

1. **Write a Python program that has a loop that prints out all the numbers in a List that are even.**

```
import sys
list=[]
even=[]
for i in range(0,int(sys.argv[1])):
        element=int(input("Enter Element :"))
        list.append(element)
print list
for i in range(0,len(list)):
        if((list[i]%2)==0):
                even.append(list[i])
print "All the numbers in a List that are even."
print even
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac1_5.py 5
Enter Element :23
Enter Element :42
Enter Element :55
Enter Element :33
Enter Element :78
[23, 42, 55, 33, 78]
All the numbers in a List that are even.
[42, 78]
```

2. Write a Python program to create a list of the months in an year starting from January. Using a loop, print the months in reverse order from December to January. Is there a function that will help you do this easily?

```
list=['Jan','Feb','Mar','Apr','May','Jun','Jul','Aug','Sep','Oct','Nov','Dec']
reverse_list=[]
print list
for i in range(len(list)-1,-1,-1):
        reverse_list.append(list[i])
print reverse_list
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac2_5.py
['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
['Dec', 'Nov', 'Oct', 'Sep', 'Aug', 'Jul', 'Jun', 'May', 'Apr', 'Mar', 'Feb', 'Jan']
```

3. Write a program that asks the user to enter 10 (positive) numbers. The program should then print the numbers in sorted order, from biggest to smallest.

```
import sys
list=[]
for i in range(0,10):
        element=int(input("Enter Element :"))
        list.append(element)
list.sort()
list.reverse()
print "Sorted List (Form biggest to smallest)"
print list
```

**************OUTPUT**************

```
 [comp@localhost]$ python Prac3_5.py
Enter Element :34
Enter Element :78
Enter Element :45
Enter Element :10
Enter Element :22
Enter Element :89
Enter Element :44
Enter Element :29
Enter Element :84
Enter Element :65
Sorted List (Form biggest to smallest)
[89, 84, 78, 65, 45, 44, 34, 29, 22, 10]
```

4. Perform a list and also a tuple assignment using Python to assign the first 4 prime numbers 2,3,5 and 7 to a list called primes. Append the 5th #prime number (11) to this list using the append() method. Print out the primes list. What difference do you observe while doing the same thing for
Lists and Tuples?

```
list=[2,3,5,7]
tuple=(2,3,5,7)
print "Prime Number List:",list
print "Prime Number Tuple:",tuple
list.append(11)
print "Appended Prime Number List:",list
tuple.append(11)
print "Appended Prime Number Tuple:",tuple
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac4_5.py
Prime Number List: [2, 3, 5, 7]
Prime Number Tuple: (2, 3, 5, 7)
Appended Prime Number List: [2, 3, 5, 7, 11]
Traceback (most recent call last):
  File "prime5.py", line 8, in <module>
tuple.append(11)
AttributeError: 'tuple' object has no attribute 'append'
```

5. Write a Python program to create a list by concatenating a given list which range goes from 1 to n.

```
n=int(input("Enter size:"))
list1=['p','q']
list2=[]
fori in range(1,n+1):
        list2.append(i)
print "Your List:",list2
```

```
list3=[]
fori in list2:
        for j in list1:
                list3.append(j+str(i))
print "Final List:",list3
```

**************OUTPUT**************

```
[comp@localhost]$ python Prac5_5.py
Enter size:5
Your List: [1, 2, 3, 4, 5]
Final List: ['p1', 'q1', 'p2', 'q2', 'p3', 'q3', 'p4', 'q4', 'p5', 'q5']
```

OpenCV Programs

1. Write a program to Read and Write Image.

```
import numpy as np
import cv2
np.set_printoptions(threshold=np.inf)
#img = cv2.imread('C:/Users/vbhaumik/Pictures/Capture.jpg',0)
img = cv2.imread('images/messi5.jpg',0)
cv2.imshow('first_image',img)
print img
k = cv2.waitKey(0) & 0xFF
if k == 27: # wait for ESC key to exit
        cv2.destroyAllWindows()
elif k == ord('s'): # wait for 's' key to save and exit
        cv2.imwrite('C:/Users/vbhaumik/Pictures/Capture.png',img)
        cv2.destroyAllWindows()
```

2. Write a program to work with Video.

```
import numpy as np
import cv2
cap = cv2.VideoCapture(0)
while(True):
        # Capture frame-by-frame
        ret, frame = cap.read()
        # Our operations on the frame come here
        #gray = cv2.cvtColor(frame, cv2.COLOR_BGR2GRAY)
        # Display the resulting frame
        cv2.imshow('frame',gray)
        if cv2.waitKey(1) & 0xFF == ord('q'):
                break
# When everything done, release the capture
cap.release()
cv2.destroyAllWindows()
```

3. **Write a program for Image blending**.

```
img1 = cv2.imread('ml.png')
img2 = cv2.imread('opencv_logo.jpg')
dst = cv2.addWeighted(img1,0.7,img2,0.3,0)
cv2.imshow('dst',dst)
cv2.waitKey(0)
cv2.destroyAllWindows()
```

4. Write a program to Inverse and Image.

```
import cv2
img2 = cv2.imread('images/circles.png')
inv= cv2.bitwise_not(img2)
cv2.imshow('inverse',inv)
```

5. **Write a program for Blue object tracking**.

```
import cv2
import numpy as np
cap = cv2.VideoCapture(0)
while(1):
# Take each frame
_, frame = cap.read()
# Convert BGR to HSV
hsv = cv2.cvtColor(frame, cv2.COLOR_BGR2HSV)
# define range of blue color in HSV
lower_blue = np.array([110,50,50])
upper_blue = np.array([130,255,255])
# Threshold the HSV image to get only blue colors
mask = cv2.inRange(hsv, lower_blue, upper_blue)
# Bitwise-AND mask and original image
res = cv2.bitwise_and(frame,frame, mask= mask)
cv2.imshow('frame',frame)
cv2.imshow('mask',mask)
cv2.imshow('res',res)
k = cv2.waitKey(5) & 0xFF
if k == 27:
break
cv2.destroyAllWindows()
```

6. **Write a program for Image Thresholding.**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('gradient.png',0)
ret,thresh1 = cv2.threshold(img,127,255,cv2.THRESH_BINARY)
ret,thresh2 = cv2.threshold(img,127,255,cv2.THRESH_BINARY_INV)
ret,thresh3 = cv2.threshold(img,127,255,cv2.THRESH_TRUNC)
ret,thresh4 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO)
```

```
ret,thresh5 = cv2.threshold(img,127,255,cv2.THRESH_TOZERO_INV)
titles = ['Original Image','BINARY','BINARY_INV','TRUNC','TOZERO','TOZERO_INV']
images = [img, thresh1, thresh2, thresh3, thresh4, thresh5]
for i in xrange(6):
plt.subplot(2,3,i+1),plt.imshow(images[i],'gray')
plt.title(titles[i])
plt.xticks([]),plt.yticks([])
plt.show()
```

7. **Write a program for Image filtering.**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('images/cameraman.tif')
kernel = np.ones((5,5),np.float32)/25
dst = cv2.filter2D(img,-1,kernel)
plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(dst),plt.title('Averaging')
plt.xticks([]), plt.yticks([])
plt.show()
```

8. **Write a program for Image blurring.**

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('opencv_logo.png')
blur = cv2.blur(img,(5,5))
plt.subplot(121),plt.imshow(img),plt.title('Original')
plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(blur),plt.title('Blurred')
plt.xticks([]), plt.yticks([])
plt.show()
```

9. **Write a program for Erosion**.

```
import cv2
import numpy as np
img = cv2.imread(‘images/morph.jpg',0)
kernel = np.ones((5,5),np.uint8)
erosion = cv2.erode(img,kernel,iterations = 1)
dilation = cv2.dilate(img,kernel,iterations = 1)
opening = cv2.morphologyEx(img, cv2.MORPH_OPEN, kernel)
closing = cv2.morphologyEx(img, cv2.MORPH_CLOSE, kernel)
gradient = cv2.morphologyEx(img, cv2.MORPH_GRADIENT, kernel)
```

10. **Write a program to find Image gradiant**.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread(‘images/blobs.png',0)
```

```
laplacian = cv2.Laplacian(img,cv2.CV_64F)
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)
plt.subplot(2,2,1),plt.imshow(img,cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,2),plt.imshow(laplacian,cmap = 'gray')
plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,3),plt.imshow(sobelx,cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,4),plt.imshow(sobely,cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
plt.show()
```

11. **Write a program for Canny Edge Detection**.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('images/blobs.png',0)
laplacian = cv2.Laplacian(img,cv2.CV_64F)
sobelx = cv2.Sobel(img,cv2.CV_64F,1,0,ksize=5)
sobely = cv2.Sobel(img,cv2.CV_64F,0,1,ksize=5)
plt.subplot(2,2,1),plt.imshow(img,cmap = 'gray')
plt.title('Original'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,2),plt.imshow(laplacian,cmap = 'gray')
plt.title('Laplacian'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,3),plt.imshow(sobelx,cmap = 'gray')
plt.title('Sobel X'), plt.xticks([]), plt.yticks([])
plt.subplot(2,2,4),plt.imshow(sobely,cmap = 'gray')
plt.title('Sobel Y'), plt.xticks([]), plt.yticks([])
plt.show()
```

12. **Write a program for Histogram Calculation**.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('home.jpg',0)
plt.hist(img.ravel(),256,[0,256]); plt.show()
```

13. **Write a program for Color Histogram**.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('home.jpg')
color = ('b','g','r')
for i,col in enumerate(color):
histr = cv2.calcHist([img],[i],None,[256],[0,256])
plt.plot(histr,color = col)
plt.xlim([0,256])
plt.show()
```

14. **Write a program for Masking**.

```
img = cv2.imread('images/autumn.tif',0)
# create a mask
mask = np.zeros(img.shape[:2], np.uint8)
mask[100:300, 100:400] = 255
masked_img = cv2.bitwise_and(img,img,mask = mask)
# Calculate histogram with mask and without mask
# Check third argument for mask
hist_full = cv2.calcHist([img],[0],None,[256],[0,256])
hist_mask = cv2.calcHist([img],[0],mask,[256],[0,256])
plt.subplot(221), plt.imshow(img, 'gray')
plt.subplot(222), plt.imshow(mask,'gray')
plt.subplot(223), plt.imshow(masked_img, 'gray')
plt.subplot(224), plt.plot(hist_full), plt.plot(hist_mask)
plt.xlim([0,256])
plt.show()
```

15. **Write a program for Histogram Equalization.**

```
img = cv2.imread('images/rice.png',0)
equ = cv2.equalizeHist(img)
res = np.hstack((img,equ)) #stacking images side-by-side
cv2.imshow("img",res)
```

16. **Write a program to find Fourier Transform**.

```
import cv2
import numpy as np
from matplotlib import pyplot as plt
img = cv2.imread('messi5.jpg',0)
f = np.fft.fft2(img)
fshift = np.fft.fftshift(f)
magnitude_spectrum = 20*np.log(np.abs(fshift))
plt.subplot(121),plt.imshow(img, cmap = 'gray')
plt.title('Input Image'), plt.xticks([]), plt.yticks([])
plt.subplot(122),plt.imshow(magnitude_spectrum, cmap = 'gray')
plt.title('Magnitude Spectrum'), plt.xticks([]), plt.yticks([])
plt.show()
```