Program that traverses directories

```
#!/usr/bin/python

import os
import pwd
import sys

dirlist = [ sys.argv[1] ]
while dirlist:
  fullpathname =  dirlist.pop()
  print fullpathname
  curlist = os.listdir(fullpathname)
  for fdname in curlist:
     newpath = fullpathname + "/" + fdname
     if os.path.isdir(newpath):
       dirlist.append(newpath)
     print newpath
```

Hash Function

- Also known as “message digest”
- Mathematical function that gives message a hash value
- The chance that two different messages will have the same message digest is statistically insignificant
- Collision occurs when multiple messages have the same hash value
- It is computationally infeasible to compute a message from its hash value or to find two messages with the same hash value
- Examples: MD5, SHA256

Program that executes a command on shell and returns output

```
#!/usr/bin/python

import subprocess

cmd       = 'shasum -a 256 file.txt'
output    = subprocess.check_output(cmd,shell=True)
print(output)
```

Python Hashlib

```
#!/usr/bin/python

import hashlib

hash_object = hashlib.sha256(b'Hello World')
hex_dig = hash_object.hexdigest()
print(hex_dig)
```

```
#!/usr/bin/python

import hashlib

mystring = raw_input('Enter String to hash: ')
hash_object = hashlib.sha256(mystring.encode())
print(hash_object.hexdigest())
```

Dictionaries (maps)

```
>>> tel = {'jack': 4098, 'sape': 4139}
>>> tel['guido'] = 4127
>>> tel
{'sape': 4139, 'guido': 4127, 'jack': 4098}
>>> tel['jack']
4098
>>> del tel['sape']
>>> tel['irv'] = 4127
>>> tel
{'guido': 4127, 'irv': 4127, 'jack': 4098}
>>> tel.keys()
['guido', 'irv', 'jack']
>>> 'guido' in tel
True
```

Iterating all (key,value) pairs of a dictionary

```
#!/usr/bin/python

tel = {'jack': 4098, 'sape': 4139}
tel['irv'] = 4127

for key, value in tel.iteritems():
   print key, value
```

```
#!/usr/bin/python3

tel = {'jack': 4098, 'sape': 4139}
tel['irv'] = 4127

for key, value in tel.items():
   print(key, value)
```

python3

Functions

```
#!/usr/bin/python

def func1(a,b):
   total = a + b
   return total

def func2(a,b):
   total = a * b
   return total

# main function
retvalue = func1( 10, 20 )
print  retvalue


retvalue = func2( 10, 20 )
print  retvalue
```

Additional notes about HW2

- Please note that testing of whether files or directories are identical is based on the hash of the contents of files or directories (and NOT on their names)
- For example, the following directories are identical.