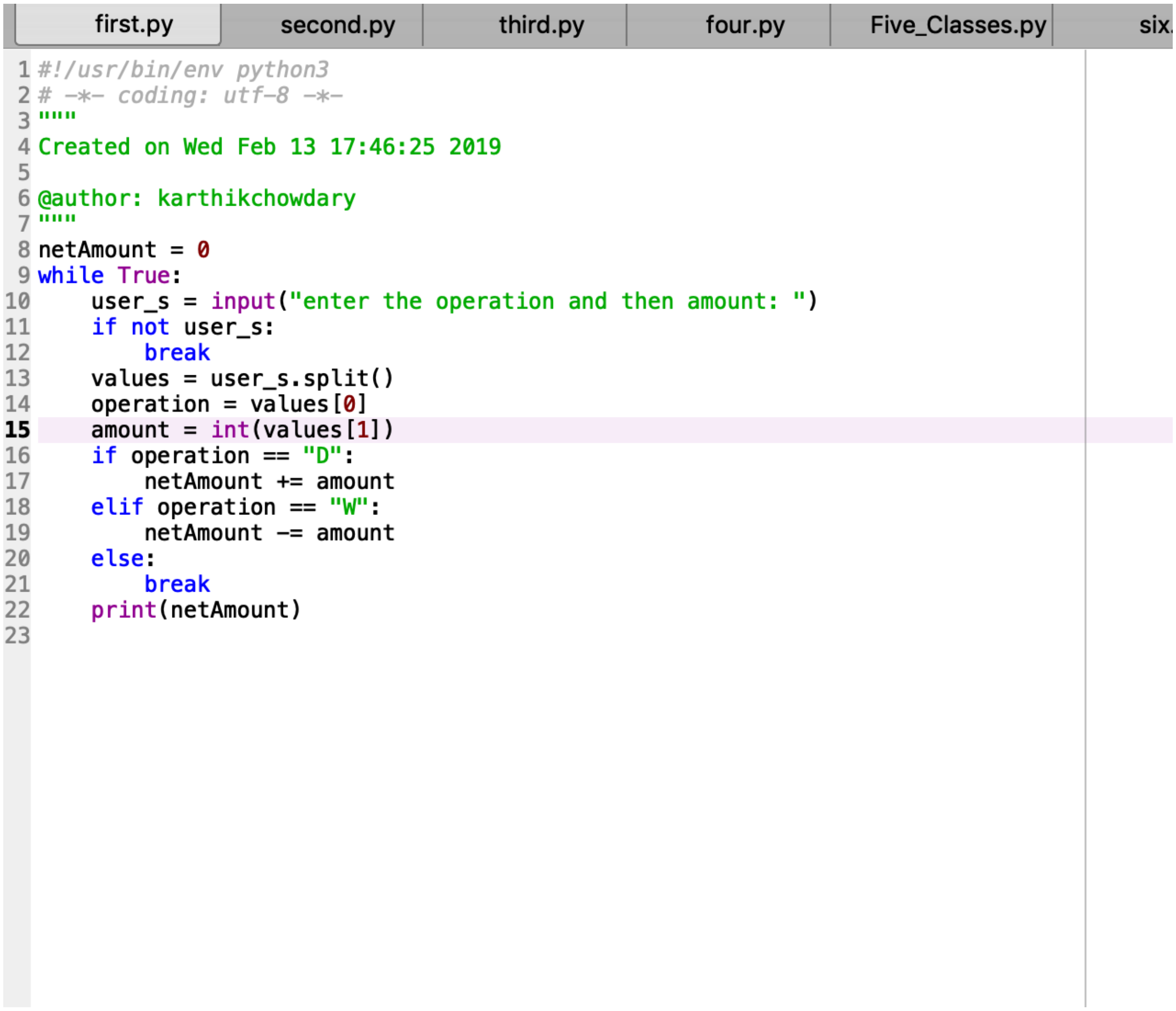
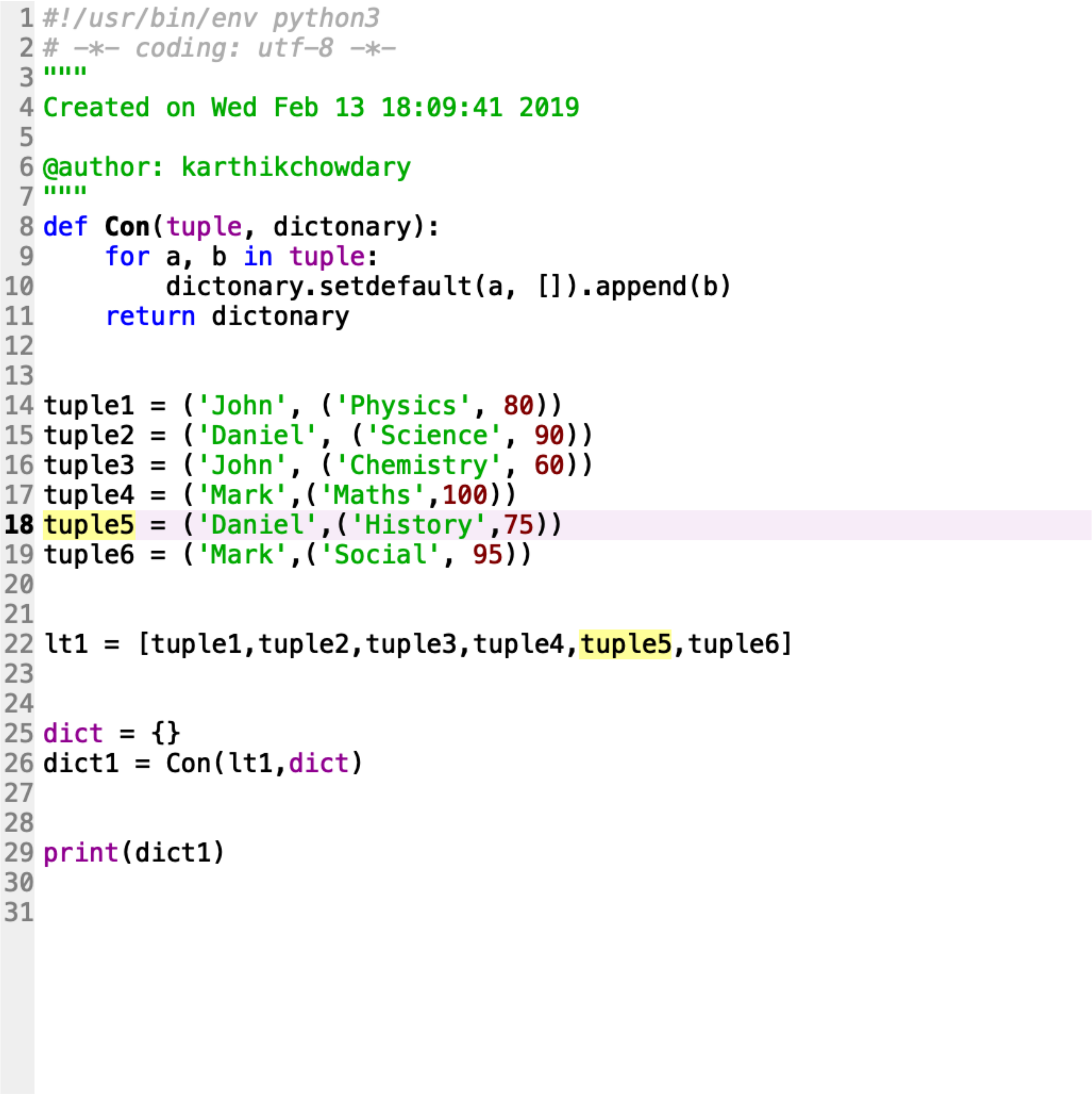
1)



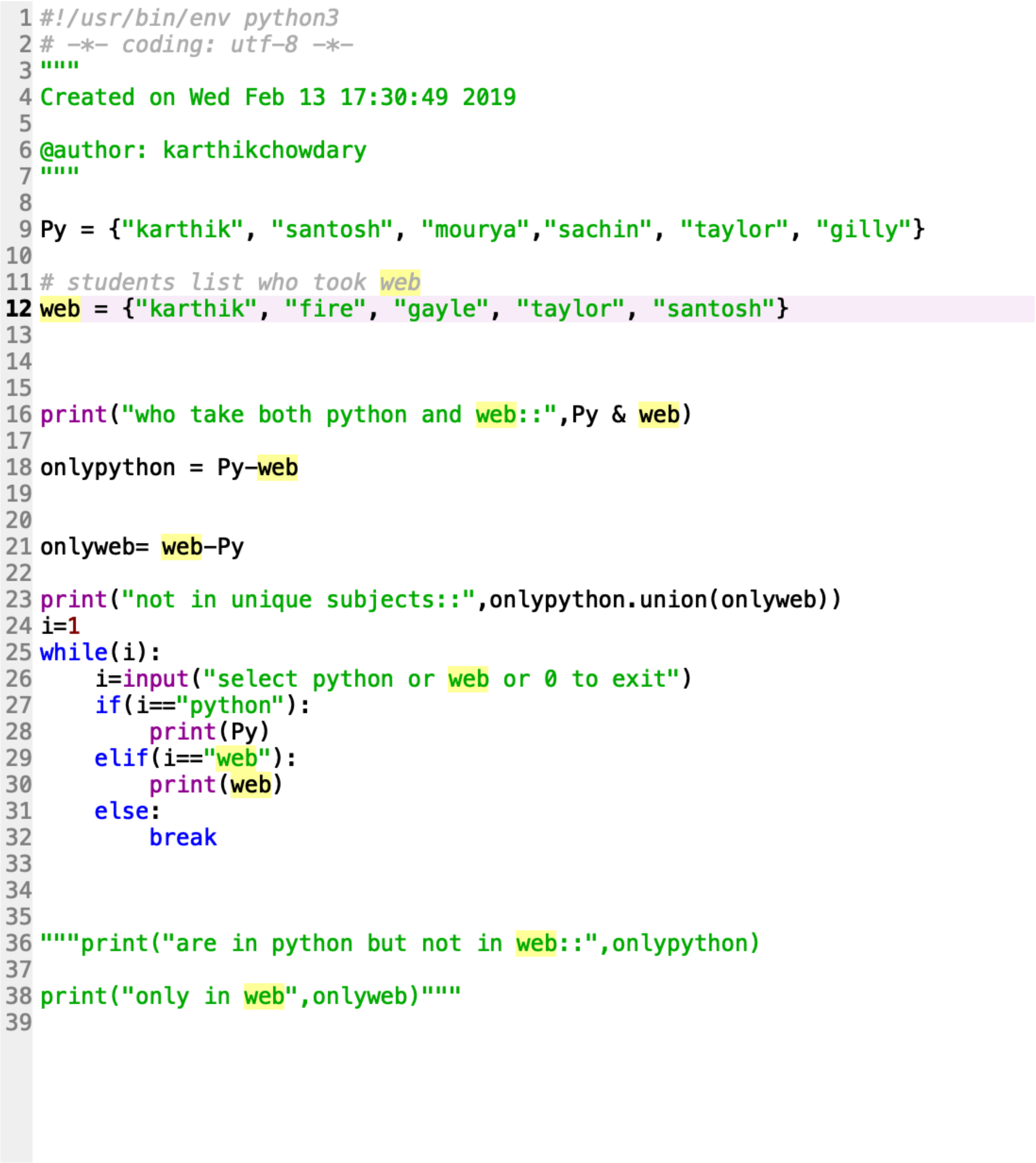
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 13 17:46:25 2019

@author: karthikchowdary
"""
netAmount = 0
while True:
    user_s = input("enter the operation and then amount: ")
    if not user_s:
        break
    values = user_s.split()
    operation = values[0]
    amount = int(values[1])
    if operation == "D":
        netAmount += amount
    elif operation == "W":
        netAmount -= amount
    else:
        break
    print(netAmount)

```

2)



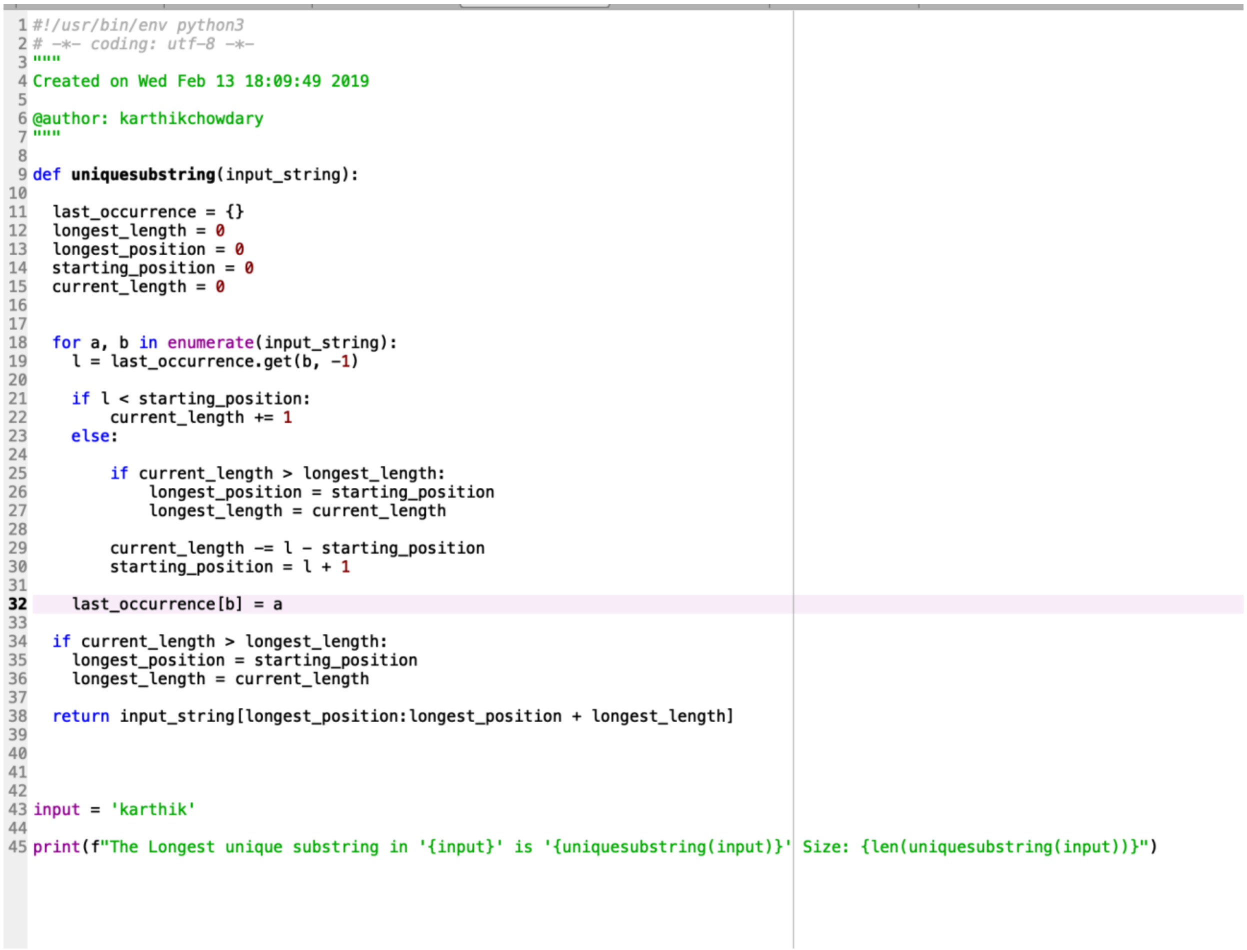
```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 13 18:09:41 2019

@author: karthikchowdary
"""
def Con(tuple, dictonary):
    for a, b in tuple:
        dictonary.setdefault(a, []).append(b)
    return dictonary


tuple1 = ('John', ('Physics', 80))
tuple2 = ('Daniel', ('Science', 90))
tuple3 = ('John', ('Chemistry', 60))
tuple4 = ('Mark',('Maths',100))
tuple5 = ('Daniel',('History',75))
tuple6 = ('Mark',('Social', 95))


lt1 = [tuple1,tuple2,tuple3,tuple4,tuple5,tuple6]


dict = {}
dict1 = Con(lt1,dict)


print(dict1)
```

3)



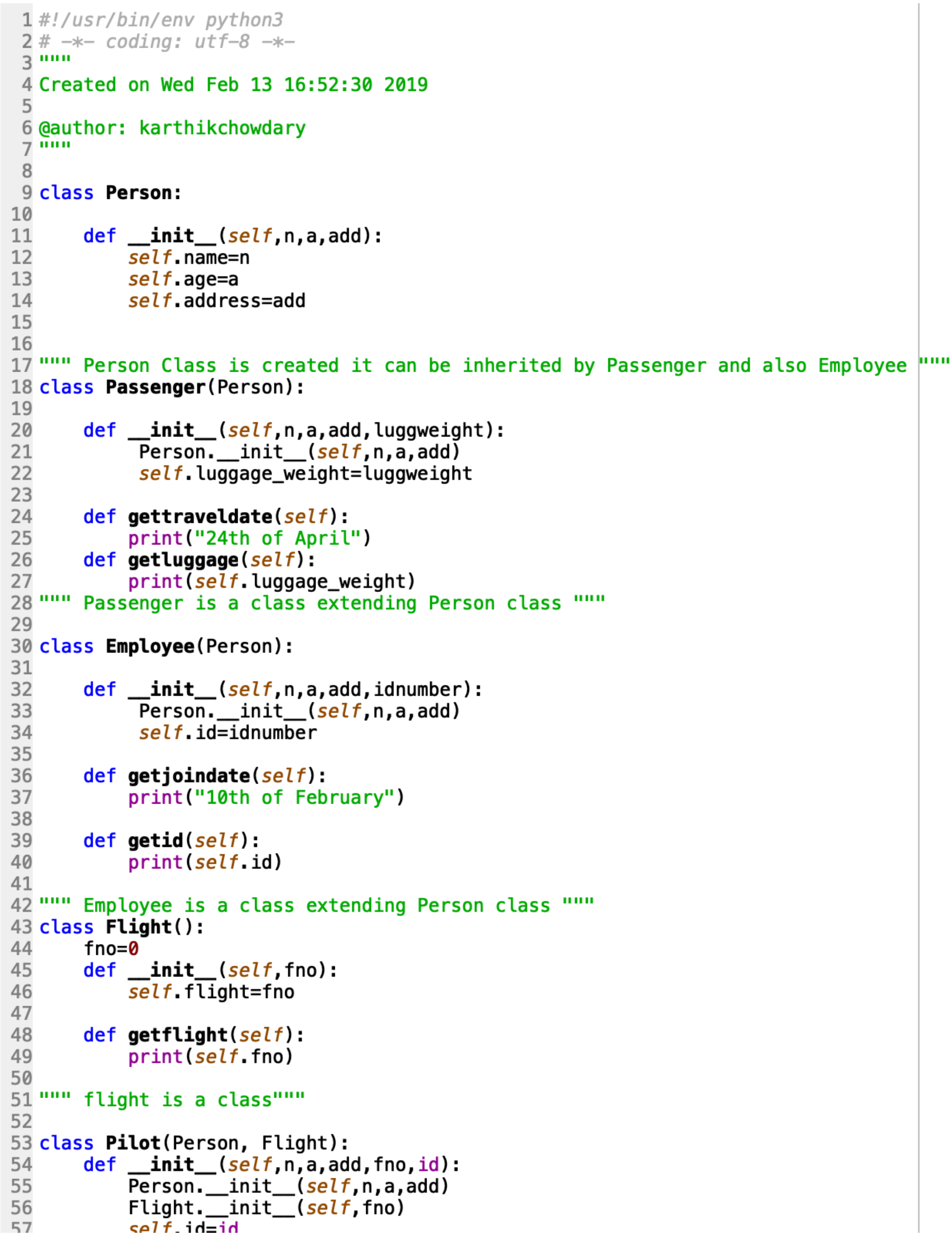
```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 13 17:30:49 2019

@author: karthikchowdary
"""

Py = {"karthik", "santosh", "mourya","sachin", "taylor", "gilly"}

# students list who took web
web = {"karthik", "fire", "gayle", "taylor", "santosh"}



print("who take both python and web::",Py & web)

onlypython = Py-web


onlyweb= web-Py

print("not in unique subjects::",onlypython.union(onlyweb))
i=1
while(i):
    i=input("select python or web or 0 to exit")
    if(i=="python"):
        print(Py)
    elif(i=="web"):
        print(web)
    else:
        break



"""print("are in python but not in web::",onlypython)

print("only in web",onlyweb)"""

```

4)

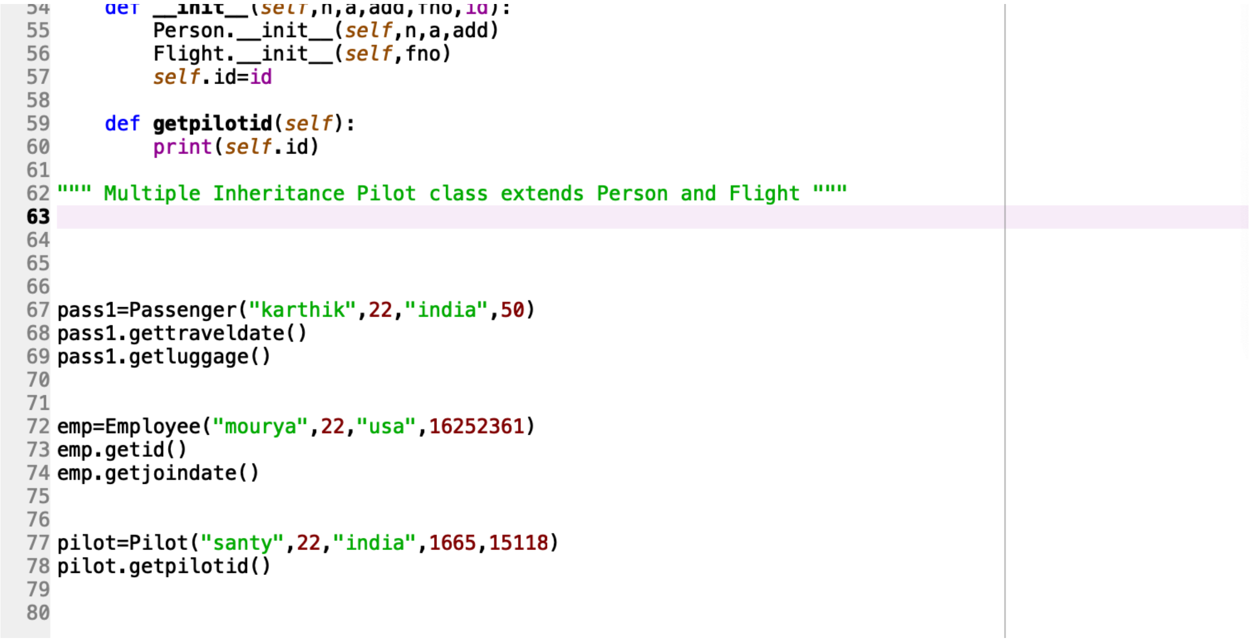


```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 13 18:09:49 2019

@author: karthikchowdary
"""

def uniquesubstring(input_string):

  last_occurrence = {}
  longest_length = 0
  longest_position = 0
  starting_position = 0
  current_length = 0


  for a, b in enumerate(input_string):
    l = last_occurrence.get(b, -1)

    if l < starting_position:
        current_length += 1
    else:

        if current_length > longest_length:
            longest_position = starting_position
            longest_length = current_length

        current_length -= l - starting_position
        starting_position = l + 1

    last_occurrence[b] = a

  if current_length > longest_length:
    longest_position = starting_position
    longest_length = current_length

  return input_string[longest_position:longest_position + longest_length]




input = 'karthik'

print(f"The Longest unique substring in '{input}' is '{uniquesubstring(input)}' Size: {len(uniquesubstring(input))}")
```

5)

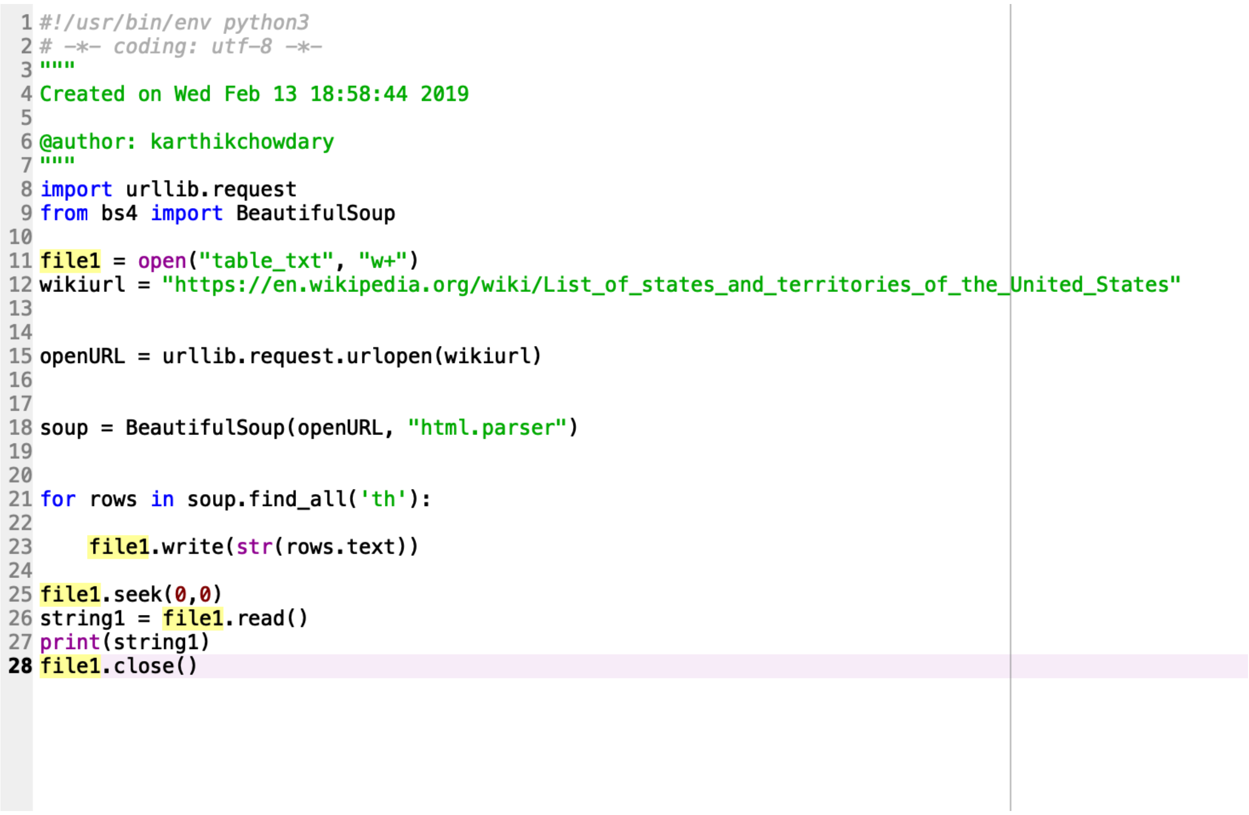


```
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 13 16:52:30 2019

@author: karthikchowdary
"""

class Person:

    def __init__(self,n,a,add):
        self.name=n
        self.age=a
        self.address=add


""" Person Class is created it can be inherited by Passenger and also Employee """
class Passenger(Person):

    def __init__(self,n,a,add,luggweight):
         Person.__init__(self,n,a,add)
         self.luggage_weight=luggweight

    def gettraveldate(self):
        print("24th of April")
    def getluggage(self):
        print(self.luggage_weight)
""" Passenger is a class extending Person class """

class Employee(Person):

    def __init__(self,n,a,add,idnumber):
         Person.__init__(self,n,a,add)
         self.id=idnumber

    def getjoindate(self):
        print("10th of February")

    def getid(self):
        print(self.id)

""" Employee is a class extending Person class """
class Flight():
    fno=0
    def __init__(self,fno):
        self.flight=fno

    def getflight(self):
        print(self.fno)

""" flight is a class"""

class Pilot(Person, Flight):
    def __init__(self,n,a,add,fno,id):
        Person.__init__(self,n,a,add)
        Flight.__init__(self,fno)
        self.id=id
```



```
    def __init__(self,n,a,add,fno,id):
        Person.__init__(self,n,a,add)
        Flight.__init__(self,fno)
        self.id=id

    def getpilotid(self):
        print(self.id)

""" Multiple Inheritance Pilot class extends Person and Flight """




pass1=Passenger("karthik",22,"india",50)
pass1.gettraveldate()
pass1.getluggage()


emp=Employee("mourya",22,"usa",16252361)
emp.getid()
emp.getjoindate()


pilot=Pilot("santy",22,"india",1665,15118)
pilot.getpilotid()


```

6)



```python
#!/usr/bin/env python3
# -*- coding: utf-8 -*-
"""
Created on Wed Feb 13 18:58:44 2019

@author: karthikchowdary
"""
import urllib.request
from bs4 import BeautifulSoup

file1 = open("table_txt", "w+")
wikiurl = "https://en.wikipedia.org/wiki/List_of_states_and_territories_of_the_United_States"


openURL = urllib.request.urlopen(wikiurl)


soup = BeautifulSoup(openURL, "html.parser")


for rows in soup.find_all('th'):

    file1.write(str(rows.text))

file1.seek(0,0)
string1 = file1.read()
print(string1)
file1.close()
```