

ARTIFICIAL INTELLIGENCE LABORATORY

LPCCS-106

Guneet Kohli

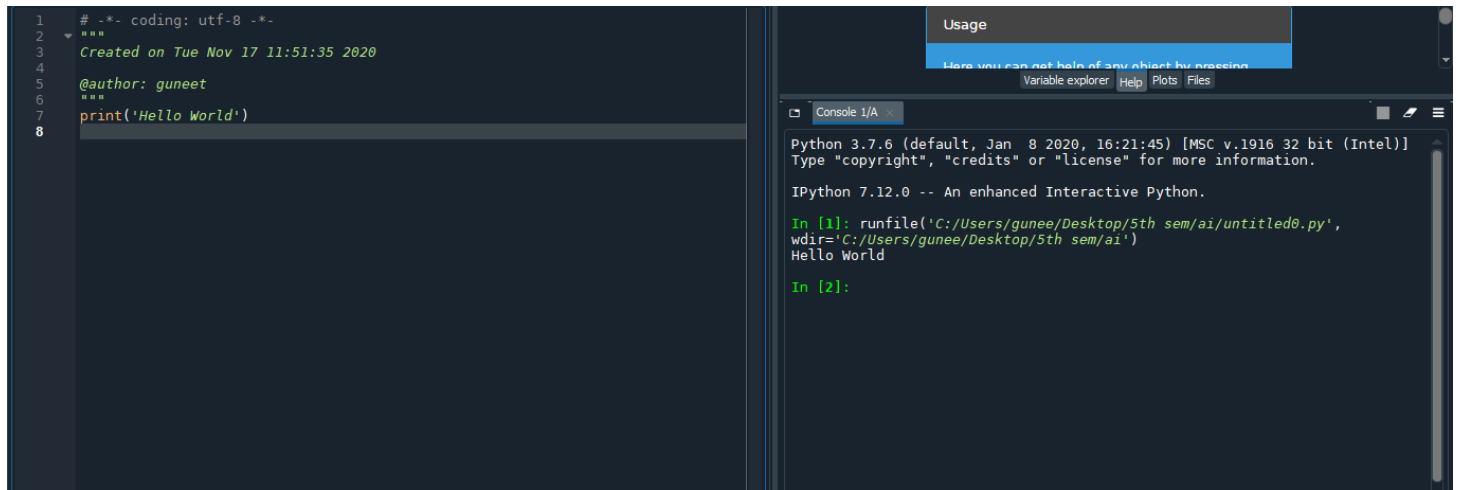
URN: 1805172 CRN: 1815017

D3 CSEA1

Serial Number	Program Name	Page Number
1	Print Hello World	2
2	Sum of 2 Numbers	3
3	Sum of 2 Complex Numbers	4
4	WAP to check whether a number is Odd or Even	5
5	WAP to find the Greatest of 3 numbers	6
6	WAP to check whether a number is prime or not	7
7	WAP to find prime numbers in a range using for loop in Python	8
8	Function that prints a simple message "I am in a function".	9
9	WAP to create a simple calculator. user will enter choice (addition/ subtraction/ division/ multiplication/ exit) specific function will be called.	10 -12
10	Pass arguments in above[9] (pass arbitrary arguments in Add function)	13
11	Implement stacks and queues using lists	14-15
12	WAP to implement A* Algorithm	16-19
13	WAP to implement trees	20-22
14	WAP to implement Water Jug Problem using DFS	23-26

1. WAP to print Hello World in Python

```
print('Hello World')
```



The screenshot displays a Python IDE interface. On the left, a code editor shows a script with the following content:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Nov 17 11:51:35 2020
4
5 @author: guneet
6 """
7 print('Hello World')
8
```

On the right, a console window titled 'Console 1/A' shows the execution output:

```
Python 3.7.6 (default, Jan 8 2020, 16:21:45) [MSC v.1916 32 bit (Intel)]
Type "copyright", "credits" or "license()" for more information.

IPython 7.12.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
Hello World
wdir='C:/Users/gunee/Desktop/5th sem/ai')

In [2]:
```

2. WAP to calculate sum of two numbers in Python

Store input numbers

```
num1 = input('Enter first number: ')
```

```
num2 = input('Enter second number: ')
```

Add two numbers

```
sum = float(num1) + float(num2)
```

Display the sum

```
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

The screenshot shows the Spyder Python IDE interface. The left pane displays the source code for a file named 'untitled0.py'. The code is as follows:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Nov 17 11:51:35 2020
4
5 @author: guneet
6 """
7
8 # Store input numbers
9 num1 = input('Enter first number: ')
10 num2 = input('Enter second number: ')
11
12 # Add two numbers
13 sum = float(num1) + float(num2)
14
15 # Display the sum
16 print('The sum of {0} and {1} is {2}'.format(num1, num2, sum))
```

The right pane shows the IPython console output. It displays the execution of the code, including the prompts for input and the resulting sum. The output is as follows:

```
Python 3.7.6 (default, Jan 8 2020, 16:21:45) [MSC v.1916 32 bit (Intel)]
Type "copyright", "credits" or "license" for more information.

IPython 7.12.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
Hello World

In [2]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
File "C:/Users/gunee/Desktop/5th sem/ai/untitled0.py", line 16
    print('The sum of {0} and {1} is {2}'.format(num1, num2, sum)
SyntaxError: unexpected EOF while parsing

In [3]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')

Enter first number: 54
Enter second number: 33
The sum of 54 and 33 is 87.0

In [4]:
```

The status bar at the bottom indicates the environment is 'conda: base (Python 3.7.6)', the cursor is at 'Line 16, Col 63', the encoding is 'UTF-8', and the line ending is 'CRLF'. A 'screenrec' watermark is visible in the bottom right corner.

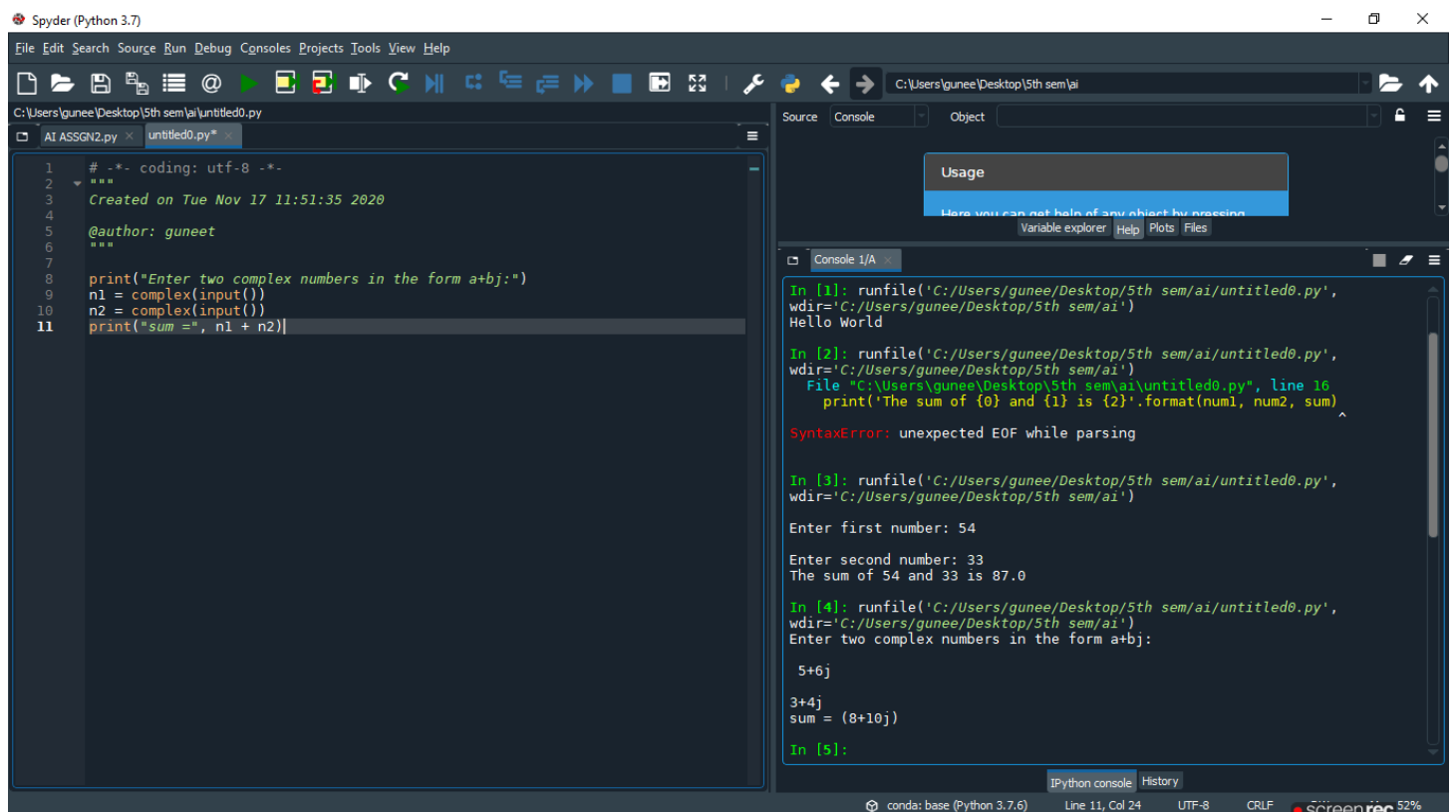
3. WAP to calculate sum of two complex numbers in Python

```
print("Enter two complex numbers in the form a+bj:")
```

```
n1 = complex(input())
```

```
n2 = complex(input())
```

```
print("sum =", n1 + n2)
```



The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named 'untitled0.py' with the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Nov 17 11:51:35 2020
4
5 @author: guneet
6 """
7
8 print("Enter two complex numbers in the form a+bj:")
9 n1 = complex(input())
10 n2 = complex(input())
11 print("sum =", n1 + n2)
```

The right pane shows the IPython console with the following output:

```
In [1]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
Hello World

In [2]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
File "C:/Users/gunee/Desktop/5th sem/ai/untitled0.py", line 16
print('The sum of {0} and {1} is {2}'.format(num1, num2, sum)
^
SyntaxError: unexpected EOF while parsing

In [3]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
Enter first number: 54
Enter second number: 33
The sum of 54 and 33 is 87.0

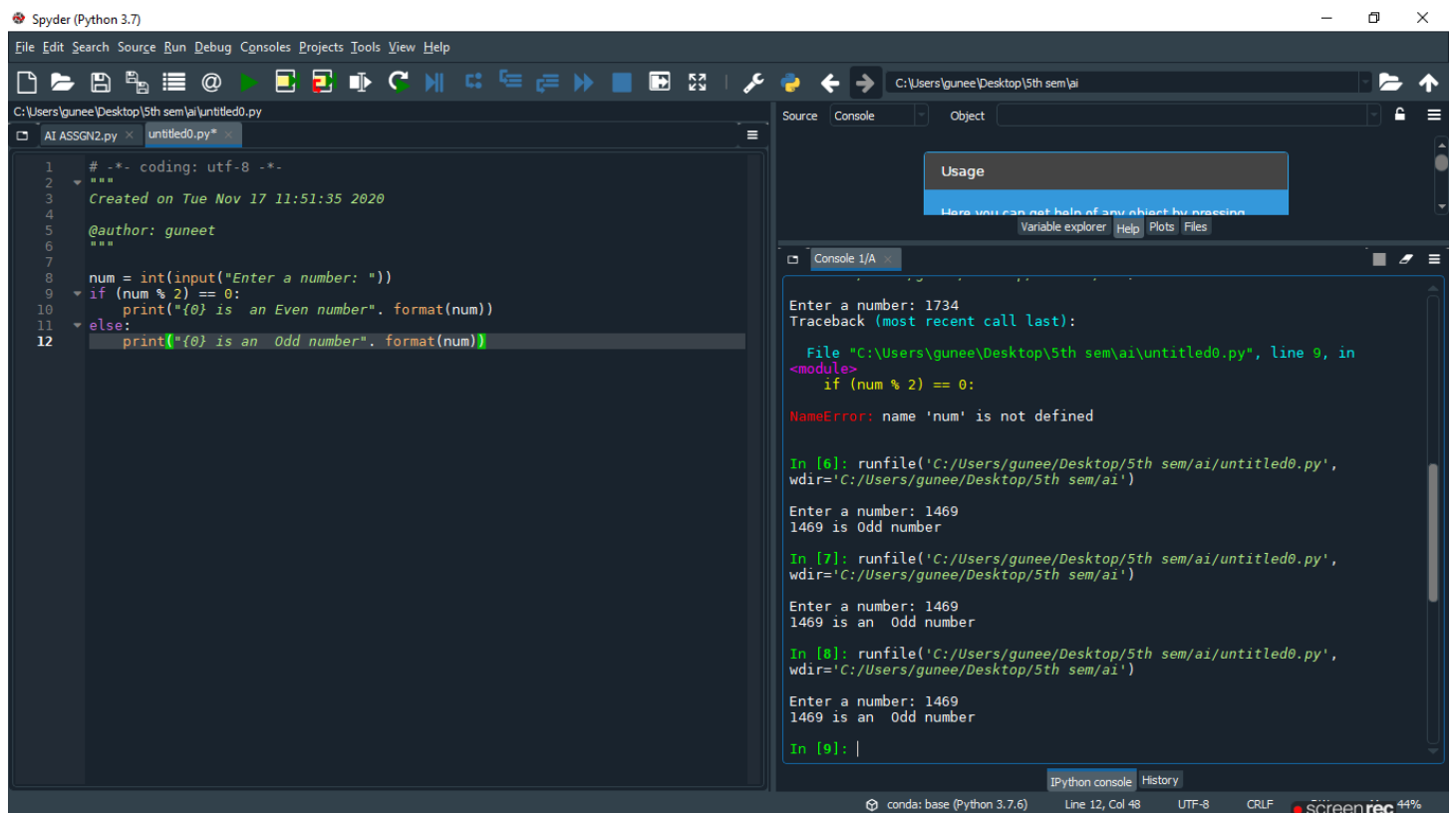
In [4]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
Enter two complex numbers in the form a+bj:
5+6j
3+4j
sum = (8+10j)

In [5]:
```

The status bar at the bottom indicates 'conda: base (Python 3.7.6)', 'Line 11, Col 24', 'UTF-8', 'CRLF', and 'screenrec 52%'.

4. WAP to check if number is even or odd in Python

```
num = int(input("Enter a number: "))  
  
if (num % 2) == 0:  
    print("{0} is an Even number".format(num))  
else:  
    print("{0} is an Odd number".format(num))
```



The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named 'untitled0.py' with the following code:

```
1 # -*- coding: utf-8 -*-  
2 """  
3 Created on Tue Nov 17 11:51:35 2020  
4  
5 @author: guneet  
6 """  
7  
8 num = int(input("Enter a number: "))  
9 if (num % 2) == 0:  
10     print("{0} is an Even number".format(num))  
11 else:  
12     print("{0} is an Odd number".format(num))
```

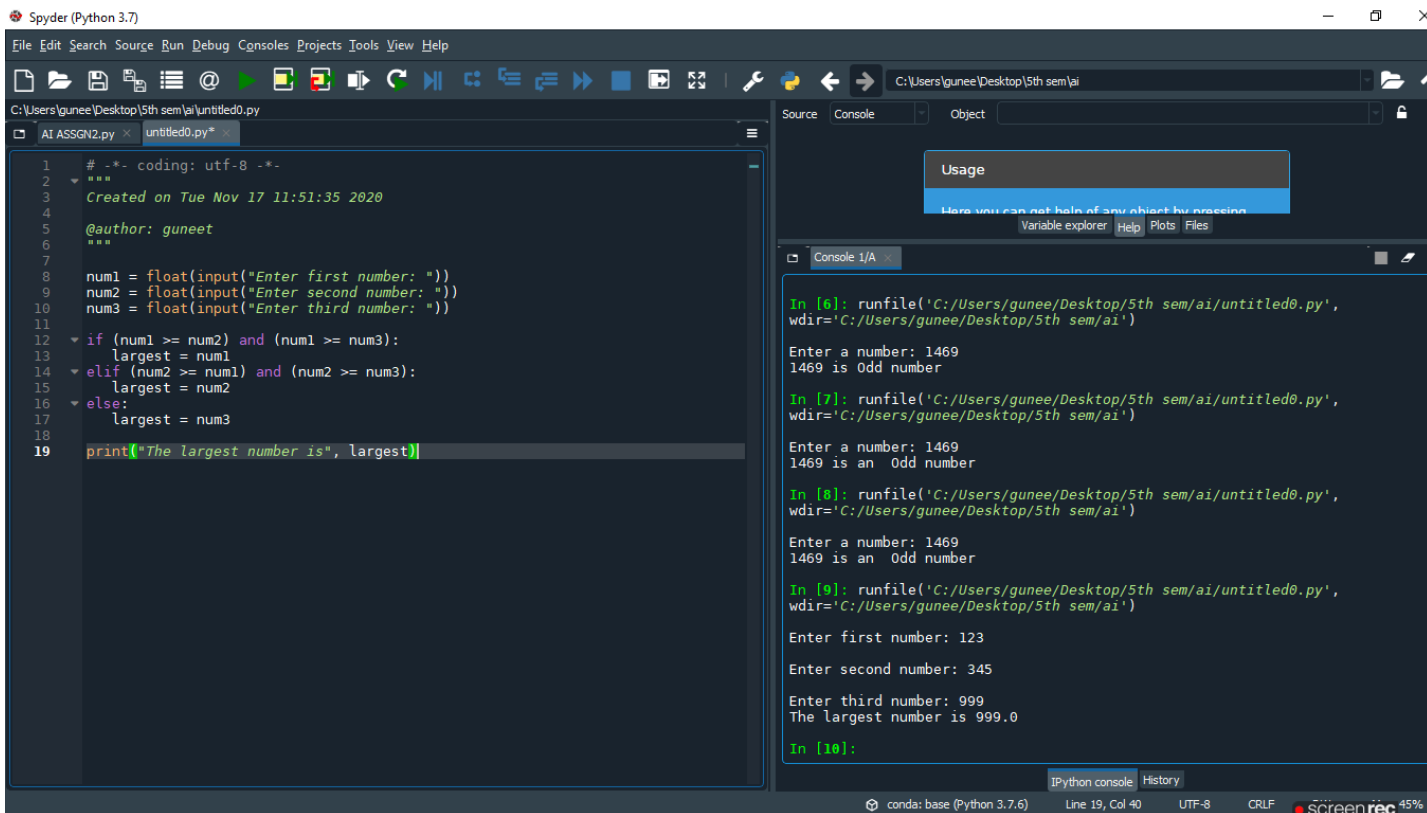
The right pane shows the IPython console with the following output:

```
Enter a number: 1734  
Traceback (most recent call last):  
  File "C:\Users\gunee\Desktop\5th sem\ai\untitled0.py", line 9, in  
<module>  
    if (num % 2) == 0:  
NameError: name 'num' is not defined  
  
In [6]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter a number: 1469  
1469 is Odd number  
  
In [7]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter a number: 1469  
1469 is an Odd number  
  
In [8]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter a number: 1469  
1469 is an Odd number  
  
In [9]: |
```

The status bar at the bottom indicates the environment is 'conda: base (Python 3.7.6)', the cursor is at 'Line 12, Col 48', the encoding is 'UTF-8', and the line endings are 'CRLF'. A 'screenrec' watermark is visible in the bottom right corner.

5. WAP to find greatest of 3 numbers in Python

```
num1 = float(input("Enter first number: "))  
num2 = float(input("Enter second number: "))  
num3 = float(input("Enter third number: "))  
  
if (num1 >= num2) and (num1 >= num3):  
    largest = num1  
elif (num2 >= num1) and (num2 >= num3):  
    largest = num2  
else:  
    largest = num3  
  
print("The largest number is", largest)
```



The screenshot shows the Spyder Python IDE interface. The left pane displays the source code for a file named 'untitled0.py'. The code is a Python script to find the largest of three numbers. The right pane shows the console output, which includes the execution of the script and the user's input for three numbers: 1469, 1469, and 123. The output shows that 1469 is the largest number and is an odd number.

```
1 # -*- coding: utf-8 -*-  
2 """  
3 Created on Tue Nov 17 11:51:35 2020  
4  
5 @author: guneet  
6 """  
7  
8 num1 = float(input("Enter first number: "))  
9 num2 = float(input("Enter second number: "))  
10 num3 = float(input("Enter third number: "))  
11  
12 if (num1 >= num2) and (num1 >= num3):  
13     largest = num1  
14 elif (num2 >= num1) and (num2 >= num3):  
15     largest = num2  
16 else:  
17     largest = num3  
18  
19 print("The largest number is", largest)
```

Console 1/A

```
In [6]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
            wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter a number: 1469  
1469 is Odd number  
  
In [7]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
            wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter a number: 1469  
1469 is an Odd number  
  
In [8]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
            wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter a number: 1469  
1469 is an Odd number  
  
In [9]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
            wdir='C:/Users/gunee/Desktop/5th sem/ai')  
Enter first number: 123  
Enter second number: 345  
Enter third number: 999  
The largest number is 999.0  
  
In [10]:
```

Python console | History

conda: base (Python 3.7.6) | Line 19, Col 40 | UTF-8 | CRLF | screenrec 45%

6. WAP to find whether a number is prime or not in Python

```
num = int(input("Enter a number: "))

# prime numbers are greater than 1

if num > 1:

    # check for factors

    for i in range(2,num):

        if (num % i) == 0:

            print(num,"is not a prime number")

            print(i,"times",num//i,"is",num)

            break

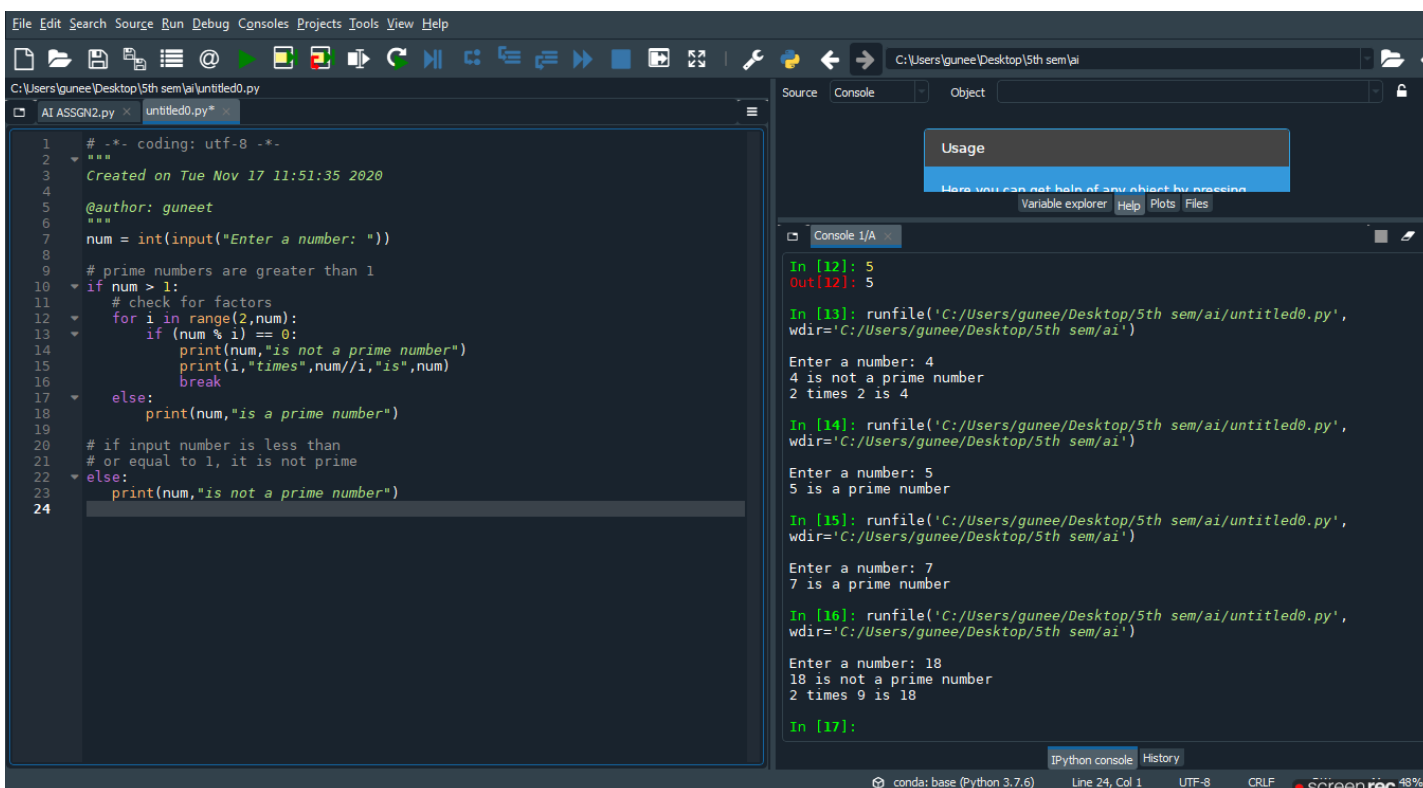
    else:

        print(num,"is a prime number")

# if input number is less than or equal to 1, it is not prime

else:

    print(num,"is not a prime number")
```



The screenshot shows a Python IDE with a dark theme. The left pane displays the source code for a file named 'untitled0.py'. The code is a Python program to check if a number is prime. It prompts the user to enter a number, then checks if it is greater than 1. If it is, it iterates from 2 to the number to find factors. If a factor is found, it prints that the number is not prime and shows the factorization. If no factors are found, it prints that the number is prime. If the input number is less than or equal to 1, it prints that the number is not prime. The right pane shows the console output, which includes the program's execution for inputs 4, 5, 7, and 18. The output shows that 4 is not prime (2 times 2), 5 is prime, 7 is prime, and 18 is not prime (2 times 9).

```
1  # -*- coding: utf-8 -*-
2
3  Created on Tue Nov 17 11:51:35 2020
4
5  @author: guneet
6
7  num = int(input("Enter a number: "))
8
9  # prime numbers are greater than 1
10 if num > 1:
11     # check for factors
12     for i in range(2,num):
13         if (num % i) == 0:
14             print(num,"is not a prime number")
15             print(i,"times",num//i,"is",num)
16             break
17     else:
18         print(num,"is a prime number")
19
20 # if input number is less than
21 # or equal to 1, it is not prime
22 else:
23     print(num,"is not a prime number")
24
```

Console 1/A

```
In [12]: 5
Out[12]: 5

In [13]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')

Enter a number: 4
4 is not a prime number
2 times 2 is 4

In [14]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')

Enter a number: 5
5 is a prime number

In [15]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')

Enter a number: 7
7 is a prime number

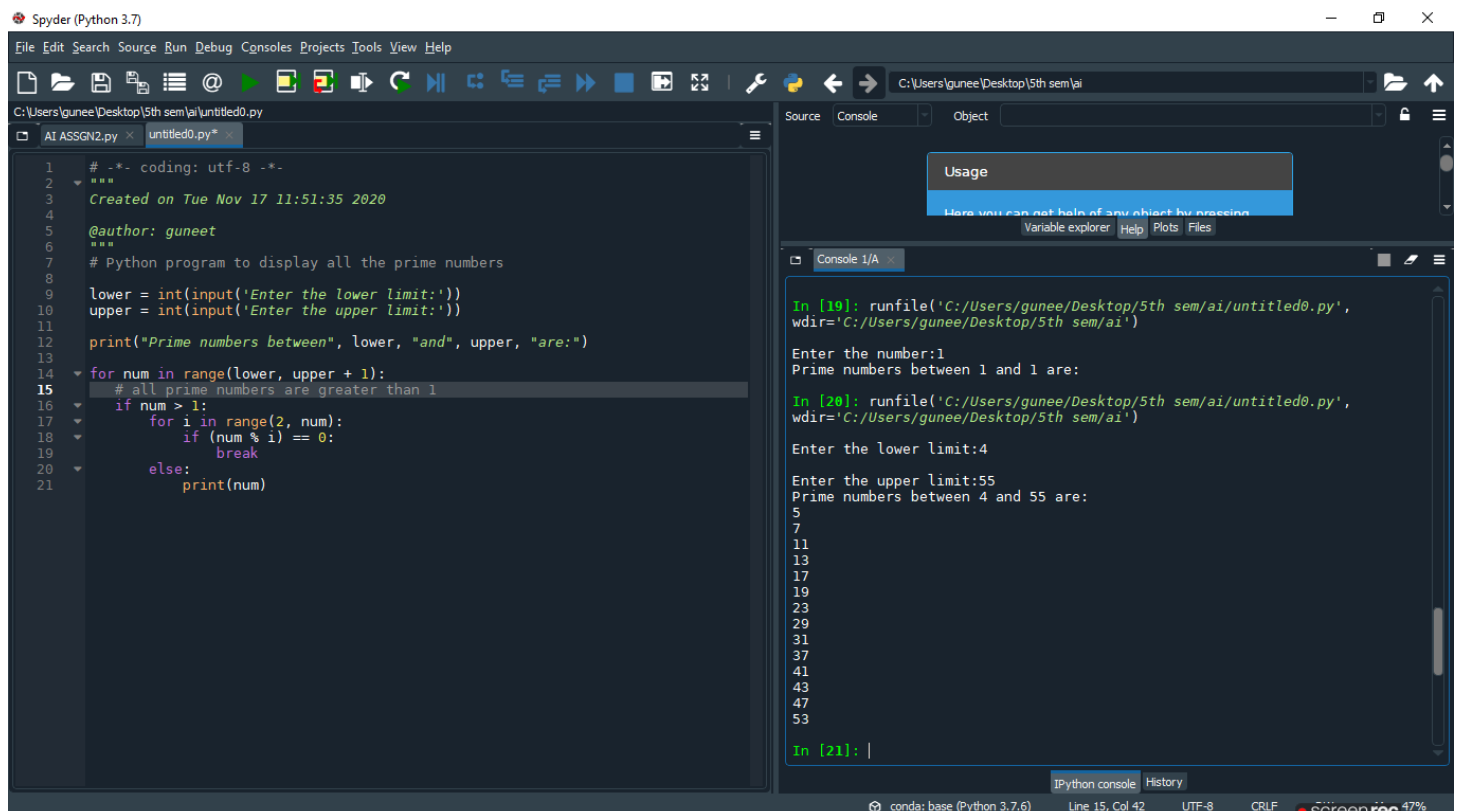
In [16]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')

Enter a number: 18
18 is not a prime number
2 times 9 is 18

In [17]:
```


7. WAP to find prime numbers in a range using for loop in Python

```
lower = int(input('Enter the lower limit:'))  
upper = int(input('Enter the upper limit:'))  
  
print("Prime numbers between", lower, "and", upper, "are:")  
  
for num in range(lower, upper + 1):  
    # all prime numbers are greater than 1  
    if num > 1:  
        for i in range(2, num):  
            if (num % i) == 0:  
                break  
        else:  
            print(num)
```



The screenshot shows the Spyder Python IDE interface. The left pane displays the source code for a program that finds prime numbers in a given range. The code is as follows:

```
1  # -*- coding: utf-8 -*-  
2  """  
3  Created on Tue Nov 17 11:51:35 2020  
4  @author: guneet  
5  """  
6  # Python program to display all the prime numbers  
7  
8  
9  lower = int(input('Enter the lower limit:'))  
10 upper = int(input('Enter the upper limit:'))  
11  
12 print("Prime numbers between", lower, "and", upper, "are:")  
13  
14 for num in range(lower, upper + 1):  
15     # all prime numbers are greater than 1  
16     if num > 1:  
17         for i in range(2, num):  
18             if (num % i) == 0:  
19                 break  
20         else:  
21             print(num)
```

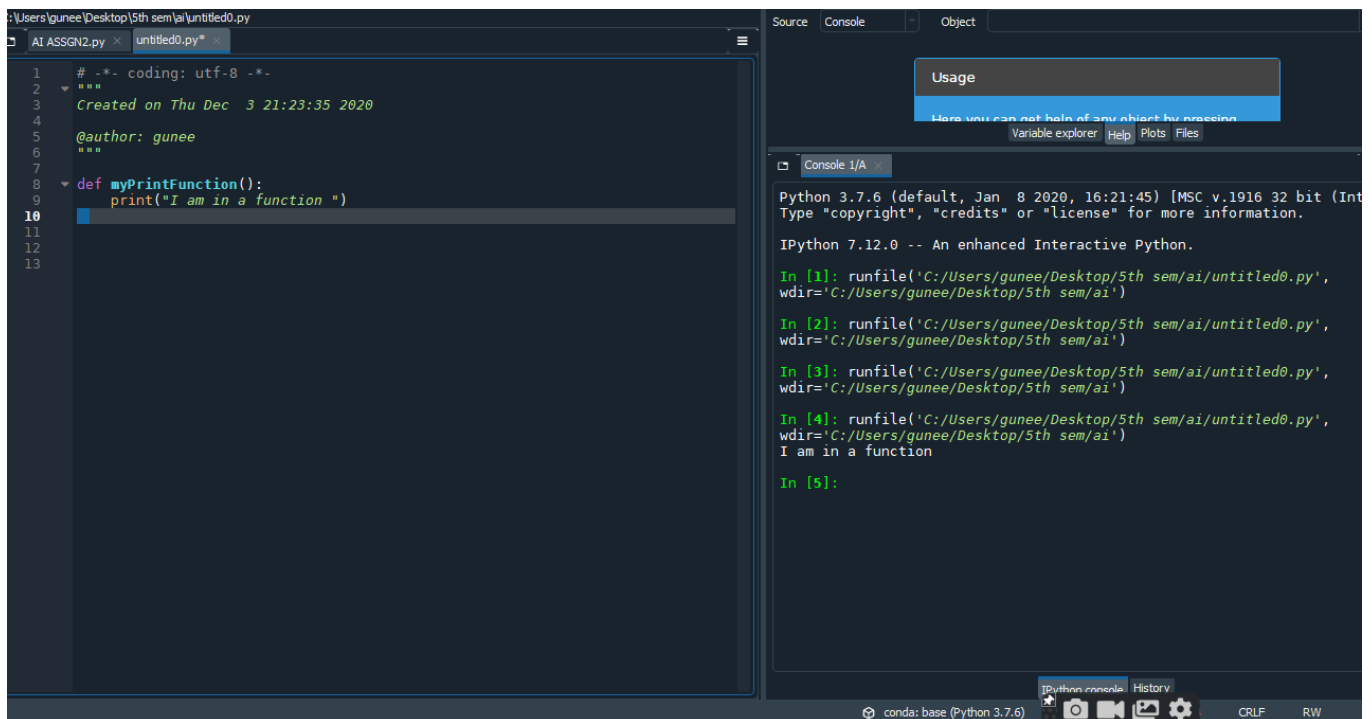
The right pane shows the IPython console with the following output:

```
In [19]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
             wdir='C:/Users/gunee/Desktop/5th sem/ai')  
  
Enter the number:1  
Prime numbers between 1 and 1 are:  
  
In [20]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',  
             wdir='C:/Users/gunee/Desktop/5th sem/ai')  
  
Enter the lower limit:4  
Enter the upper limit:55  
Prime numbers between 4 and 55 are:  
5  
7  
11  
13  
17  
19  
23  
29  
31  
37  
41  
43  
47  
53  
  
In [21]: |
```

The status bar at the bottom indicates the environment is 'conda: base (Python 3.7.6)', the current line is 15, column 42, and the encoding is UTF-8. A 'screenrec' watermark is visible in the bottom right corner.

8. Function that prints a simple message “I am in a function”.

```
def myPrintFunction():  
    print("I am in a function ")  
  
myPrintFunction()
```



The screenshot displays a Jupyter Notebook environment. The left pane shows a Python script with the following code:

```
1 # -*- coding: utf-8 -*-  
2  
3 Created on Thu Dec 3 21:23:35 2020  
4  
5 @author: guneet  
6  
7  
8 def myPrintFunction():  
9     print("I am in a function ")  
10  
11  
12  
13
```

The right pane shows the console output, which includes the IPython version and the results of running the script multiple times:

```
Python 3.7.6 (default, Jan 8 2020, 16:21:45) [MSC v.1916 32 bit (Intel)] Type "copyright", "credits" or "license()" for more information.  
IPython 7.12.0 -- An enhanced Interactive Python.  
  
In [1]: runfile('C:/Users/guneet/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/guneet/Desktop/5th sem/ai')  
  
In [2]: runfile('C:/Users/guneet/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/guneet/Desktop/5th sem/ai')  
  
In [3]: runfile('C:/Users/guneet/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/guneet/Desktop/5th sem/ai')  
  
In [4]: runfile('C:/Users/guneet/Desktop/5th sem/ai/untitled0.py',  
wdir='C:/Users/guneet/Desktop/5th sem/ai')  
I am in a function  
  
In [5]:
```

The status bar at the bottom indicates the environment is 'conda: base (Python 3.7.6)'.

9. WAP to create a simple calculator. user will enter choice (addition/ subtraction/ division/ multiplication/ exit) specific function will be called.

```
print("Name: Guneet Kohli \nURN: 1805172")
```

```
def add():
```

```
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
```

```
    print(num1+num2)
```

```
def add():
```

```
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
```

```
    print(num1+num2)
```

```
def sub():
```

```
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
```

```
    print(num1-num2)
```

```
def multiply():
```

```
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
```

```
    print(num1*num2)
```

```
def divide():
```

```
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
```

```
    print(num1/num2)
```

```
x = True
```

```
print("Please select operation\n 1. Add\n 2. Subtract\n 3. Multiply\n 4. Divide\n 5. Exit")
```

```
# Take input from the user
```

```
while x:
```

```
    select = int(input("Select operations form 1, 2, 3, 4, 5 :"))
```

```
    if select == 1:
```

```
        add()
```

```
    elif select == 2:
```

```
        sub()
```

```
    elif select == 3:
```

```
        multiply()
```

```
    elif select == 4:
```

```
        divide()
```

```
    elif select == 5:
```

```
        print("Thanks For Using.")
```

```
        x = False
```

```
    else:
```

```
        print("Invalid input")
```

```
@author: guneet
"""
print("Name: Guneet Kohli \nURN: 1805172")
def add():
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
    print(num1+num2)
def add():
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
    print(num1+num2)
def sub():
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
    print(num1-num2)
def multiply():
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
    print(num1*num2)
def divide():
    num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
    print(num1/num2)

x = True

print("Please select operation\n 1. Add\n 2. Subtract\n 3. Multiply\n 4. Divide\n 5. Exit")
# Take input from the user
while x:
    select = int(input("Select operations form 1, 2, 3, 4, 5 :"))
    if select == 1:
        add()
    elif select == 2:
        sub()
    elif select == 3:
        multiply()
    elif select == 4:
        divide()
    elif select == 5:
        x = False
    else:
        print("Invalid input")
```

Console 1/A

```

C:\Users\guneet\Desktop>python 3em/a1.py
Name: Guneet Kohli
URN: 1805172
Please select operation
1. Add
2. Subtract
3. Multiply
4. Divide
5. Exit

Select operations form 1, 2, 3, 4, 5 :1

Enter 1st Number: 789999

Enter 2st Number: 345890
1135889

Select operations form 1, 2, 3, 4, 5 :3

Enter 1st Number: 34

Enter 2st Number: 5
170

Select operations form 1, 2, 3, 4, 5 :2

Enter 1st Number: 22

Enter 2st Number: 77
-55

Select operations form 1, 2, 3, 4, 5 :
```

```

5 @author: guneet
6 """
7 print("Name: Guneet Kohli \nURN: 1805172")
8 def add():
9     num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
10    print(num1+num2)
11
12 def add():
13     num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
14    print(num1+num2)
15
16 def sub():
17     num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
18    print(num1-num2)
19
20 def multiply():
21     num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
22    print(num1*num2)
23
24
25 def divide():
26     num1,num2 = int(input("Enter 1st Number: ")),int(input("Enter 2st Number: "))
27    print(num1/num2)
28
29 x = True
30
31 print("Please select operation\n 1. Add\n 2. Subtract\n 3. Multiply\n 4. Divide\n 5. Exit")
32 # Take input from the user
33
34
35 while x:
36     select = int(input("Select operations form 1, 2, 3, 4, 5 :"))
37     if select == 1:
38         add()
39     elif select == 2:
40         sub()
41     elif select == 3:
42         multiply()
43     elif select == 4:
44         divide()
45     elif select == 5:
46         x = False
47     else:
48         print("Invalid input")
```

Console 1/A

```

Select operations form 1, 2, 3, 4, 5 :1

Enter 1st Number: 789999

Enter 2st Number: 345890
1135889

Select operations form 1, 2, 3, 4, 5 :3

Enter 1st Number: 34

Enter 2st Number: 5
170

Select operations form 1, 2, 3, 4, 5 :2

Enter 1st Number: 22

Enter 2st Number: 77
-55

Select operations form 1, 2, 3, 4, 5 :4

Enter 1st Number: 7077

Enter 2st Number: 3
2359.0

Select operations form 1, 2, 3, 4, 5 :5
Thanks For Using.

In [6]:
```

10. Pass arguments in above (pass arbitrary arguments in Add function)

```
def add(numb1=0,numb2=0):
```

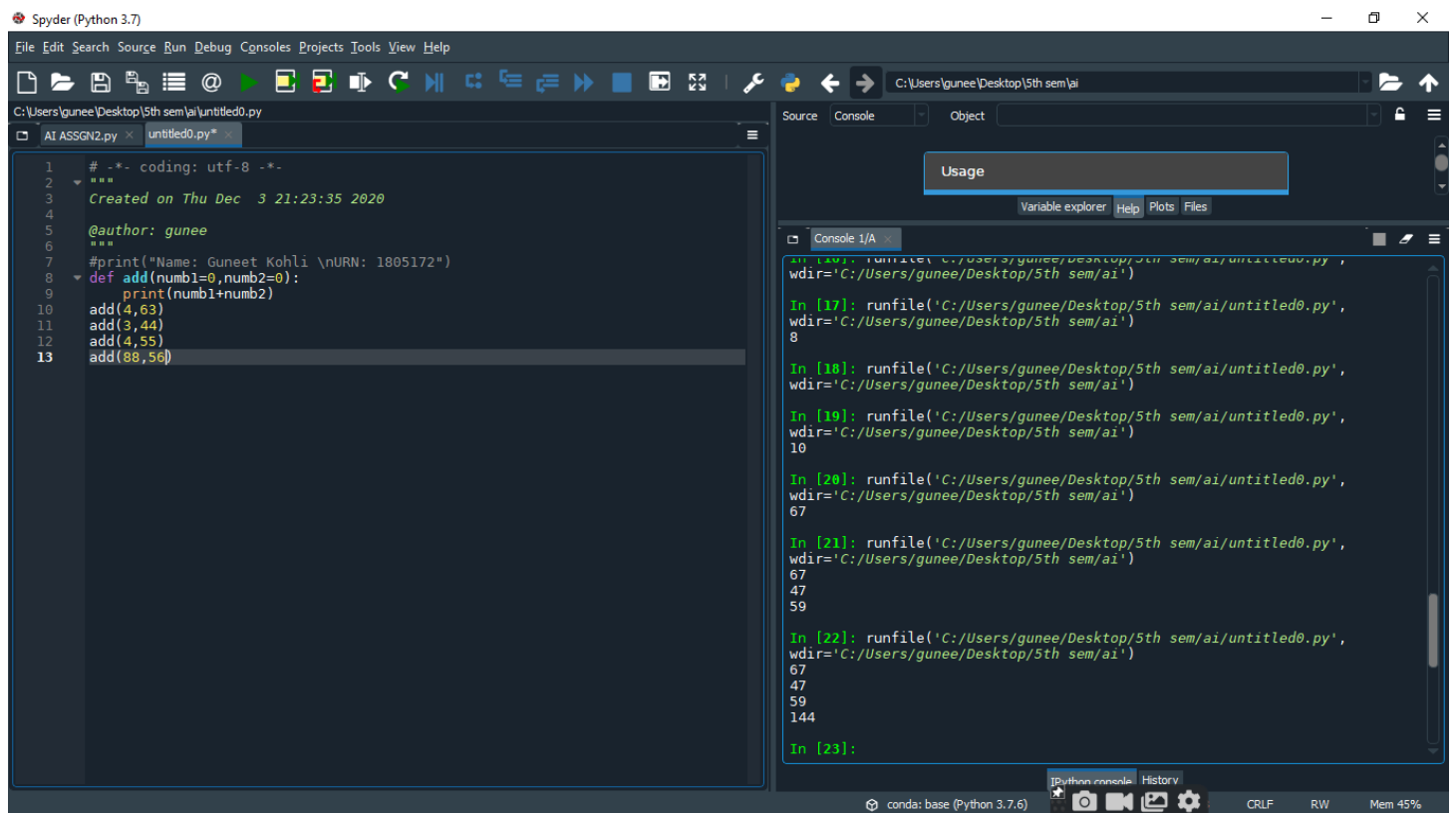
```
    print(numb1+numb2)
```

```
add(4,63)
```

```
add(3,44)
```

```
add(4,55)
```

```
add(88,56)
```



The screenshot shows the Spyder Python IDE interface. The left pane displays a Python script named 'untitled0.py' with the following code:

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Thu Dec 3 21:23:35 2020
4
5 @author: guneet
6 """
7 #print("Name: Guneet Kohli \nURN: 1805172")
8 def add(numb1=0,numb2=0):
9     print(numb1+numb2)
10 add(4,63)
11 add(3,44)
12 add(4,55)
13 add(88,56)
```

The right pane shows the console output, which displays the results of running the script multiple times. The output for each run is as follows:

```
In [17]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
8
In [18]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
10
In [19]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
10
In [20]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
67
In [21]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
67
47
59
In [22]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
67
47
59
144
In [23]:
```

11. Implement stacks and queues using lists

```
class Stack:
```

```
    def __init__(self):
```

```
        self.stack = []
```

```
    def remove(self):
```

```
        self.stack.pop(0)
```

```
    def append(self, val):
```

```
        self.stack.append(val)
```

```
    def print_stack(self):
```

```
        print(self.stack)
```

```
s = Stack()
```

```
s.append(82)
```

```
s.append(23)
```

```
s.append(11)
```

```
s.print_stack()
```

```
s.remove()
```

```
s.print_stack()
```

```
7 print("Name: Guneet Kohli \nURN: 1805172")
8 class Stack:
9     def __init__(self):
10         self.stack = []
11
12     def remove(self):
13         self.stack.pop(0)
14
15     def append(self, val):
16         self.stack.append(val)
17
18     def print_stack(self):
19         print(self.stack)
20
21 s = Stack()
22 s.append(82)
23 s.append(23)
24 s.append(11)
25 s.print_stack()
26 s.remove()
27 s.print_stack()
28
```

```
In [25]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
Name: Guneet Kohli
URN: 1805172
[82, 23, 11]
[23, 11]

In [26]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
Name: Guneet Kohli
URN: 1805172
[82, 23, 11]
[23, 11]

In [27]:
```

```
class Queue:

    def __init__(self):
        self.queue = []

    def dequeue(self):
        self.queue.pop()

    def enqueue(self, val):
        self.queue.append(val)

    def print_queue(self):
        print(self.queue)

q = Queue()

q.enqueue(113)
q.enqueue(232)
q.enqueue(443)
q.enqueue(855)

q.print_queue()

q.dequeue()
q.dequeue()
q.dequeue()

q.print_queue()
```

```
6
7 print("Name: Guneet Kohli \nURN: 1805172")
8 class Queue:
9     def __init__(self):
10         self.queue = []
11
12     def dequeue(self):
13         self.queue.pop()
14
15     def enqueue(self, val):
16         self.queue.append(val)
17
18     def print_queue(self):
19         print(self.queue)
20 q = Queue()
21
22 q.enqueue(113)
23 q.enqueue(232)
24 q.enqueue(443)
25 q.enqueue(855)
26
27 q.print_queue()
28
29 q.dequeue()
30 q.dequeue()
31 q.dequeue()
32 q.print_queue()
33
```

```
In [29]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py',
wdir='C:/Users/gunee/Desktop/5th sem/ai')
Name: Guneet Kohli
URN: 1805172
[113, 232, 443, 855]
[113]
```


12. WAP to implement A* algorithm (8 Puzzle Problem)

''''

Created on Wed Sep 11 21:08:24 2020

@author: guneet

''''

```
from copy import deepcopy
```

```
from colorama import Fore, Back, Style
```

```
# unicode
```

```
left_down_angle = '\u2514'
```

```
right_down_angle = '\u2518'
```

```
right_up_angle = '\u2510'
```

```
left_up_angle = '\u250C'
```

```
middle_junction = '\u253C'
```

```
top_junction = '\u252C'
```

```
bottom_junction = '\u2534'
```

```
right_junction = '\u2524'
```

```
left_junction = '\u251C'
```

```
bar = Style.BRIGHT + Fore.CYAN + '\u2502' + Fore.RESET + Style.RESET_ALL
```

```
dash = '\u2500'
```

```
first_line = Style.BRIGHT + Fore.CYAN + left_up_angle + dash + dash + dash + top_junction + dash + dash + dash +  
top_junction + dash + dash + dash + right_up_angle + Fore.RESET + Style.RESET_ALL
```

```
middle_line = Style.BRIGHT + Fore.CYAN + left_junction + dash + dash + dash + middle_junction + dash + dash + dash +  
middle_junction + dash + dash + dash + right_junction + Fore.RESET + Style.RESET_ALL
```

```
last_line = Style.BRIGHT + Fore.CYAN + left_down_angle + dash + dash + dash + bottom_junction + dash + dash + dash +  
bottom_junction + dash + dash + dash + right_down_angle + Fore.RESET + Style.RESET_ALL
```

```
DIRECTIONS = {"U": [-1, 0], "D": [1, 0], "L": [0, -1], "R": [0, 1]}
```

```
END = [[1, 2, 3], [4, 5, 6], [7, 8, 0]]
```

```
def print_puzzle(array):
```

```
    print(first_line)
```

```
    for a in range(len(array)):
```

```
        for i in array[a]:
```

```
    if i == 0:
        print(bar, Back.RED + ' ' + Back.RESET, end=' ')
    else:
        print(bar, i, end=' ')
print(bar)
if a == 2:
    print(last_line)
else:
    print(middle_line)
class Node:
    def __init__(self, current_node, previous_node, g, h, dir):
        self.current_node = current_node
        self.previous_node = previous_node
        self.g = g
        self.h = h
        self.dir = dir
    def f(self):
        return self.g + self.h
def get_pos(current_state, element):
    for row in range(len(current_state)):
        if element in current_state[row]:
            return (row, current_state[row].index(element))
def euclidianCost(current_state):
    cost = 0
    for row in range(len(current_state)):
        for col in range(len(current_state[0])):
            pos = get_pos(END, current_state[row][col])
            cost += abs(row - pos[0]) + abs(col - pos[1])
    return cost
def getAdjNode(node):
    listNode = []
    emptyPos = get_pos(node.current_node, 0)
    for dir in DIRECTIONS.keys():
```

```
newPos = (emptyPos[0] + DIRECTIONS[dir][0], emptyPos[1] + DIRECTIONS[dir][1])

if 0 <= newPos[0] < len(node.current_node) and 0 <= newPos[1] < len(node.current_node[0]):
    newState = deepcopy(node.current_node)
    newState[emptyPos[0]][emptyPos[1]] = node.current_node[newPos[0]][newPos[1]]
    newState[newPos[0]][newPos[1]] = 0
    # listNode += [Node(newState, node.current_node, node.g + 1, euclidianCost(newState), dir)]
    listNode.append(Node(newState, node.current_node, node.g + 1, euclidianCost(newState), dir))

return listNode

def getBestNode(openSet):
    firstIter = True
    for node in openSet.values():
        if firstIter or node.f() < bestF:
            firstIter = False
            bestNode = node
            bestF = bestNode.f()
    return bestNode

def buildPath(closedSet):
    node = closedSet[str(END)]
    branch = list()
    while node.dir:
        branch.append({
            'dir': node.dir,
            'node': node.current_node
        })
        node = closedSet[str(node.previous_node)]
    branch.append({
        'dir': "",
        'node': node.current_node
    })
    branch.reverse()
    return branch

def main(puzzle):
    open_set = {str(puzzle): Node(puzzle, puzzle, 0, euclidianCost(puzzle), "")}
    closed_set = {}
    while True:
```

```
test_node = getBestNode(open_set)

closed_set[str(test_node.current_node)] = test_node

if test_node.current_node == END:
    return buildPath(closed_set)

adj_node = getAdjNode(test_node)

for node in adj_node:
    if str(node.current_node) in closed_set.keys() or str(node.current_node) in open_set.keys() and open_set[
        str(node.current_node)].f() < node.f():
        continue

    open_set[str(node.current_node)] = node
del open_set[str(test_node.current_node)]

if __name__ == '__main__':
    br = main([[6, 2, 8],
               [4, 7, 1],
               [0, 3, 5]])

    print('total steps : ', len(br) - 1)

    print()

    print(dash + dash + right_justification, "INPUT", left_justification + dash + dash)

    for b in br:
        if b['dir'] != '':
            letter = ''

            if b['dir'] == 'U':
                letter = 'UP'

            elif b['dir'] == 'R':
                letter = "RIGHT"

            elif b['dir'] == 'L':
                letter = 'LEFT'

            elif b['dir'] == 'D':
                letter = 'DOWN'

            print(dash + dash + right_justification, letter, left_justification + dash + dash)

    print_puzzle(b['node'])

    print()

    print(dash + dash + right_justification, 'ABOVE IS THE OUTPUT', left_justification + dash + dash)
```

#OUTPUT:

```
Console I/A
In [1]: runfile('C:/Users/gunee/Desktop/5th sem/ai/AI ASSGN2.py', wdir='C:/Users/gunee/Desktop/5th sem/ai')
total steps : 24

--| INPUT |--
6 2 8
4 7 1
3 5

--| RIGHT |--
6 2 8
4 7 1
3 5

--| UP |--
6 2 8
4 1
3 7 5
```

RIGHT

6 2 8
4 1
3 7 5

UP

6 2
4 1 8
3 7 5

LEFT

6
4 1 8
3 7 5

LEFT

6 2
4 1 8
3 7 5

DOWN

4 6 2
3 1 8
7 5

DOWN

4 6 2
3 1 8
7 5

RIGHT

4 6 2
3 1 8
7 5

RIGHT

4 6 2
3 1 8
7 5

UP

4 6 2
3 1
7 5 8

UP

4 6
3 1 2
7 5 8

LEFT

4 6
3 1 2
7 5 8

DOWN

4 1 6
3 2
7 5 8

DOWN

4 1 6
3 2
7 5 8

LEFT

4 1 6
3 2
7 5 8

UP

1 6
4 3 2
7 5 8

RIGHT

1 6
4 3 2
7 5 8

DOWN

1 3 6
4 2
7 5 8

RIGHT

1 3 6
4 2
7 5 8

UP

1 3
4 2 6
7 5 8

LEFT

1 3
4 2 6
7 5 8

DOWN

1 2 3
4 6
7 5 8

DOWN

1 2 3
4 5 6
7 8

RIGHT

1 2 3
4 5 6
7 8

ABOVE IS THE OUTPUT

13. WAP to implement Trees

```
class Node(object):
    def __init__(self, value):
        self.left=None
        self.right=None
        self.value=value

class BinaryTree(object):
    def __init__(self, root):
        self.root=Node(root)

    def printTree(self, traversalType):
        if traversalType=='preorder':
            return self.PreOrderPrint(tree.root,"")

    def PreOrderPrint(self,start, traversal):
        #Root-> Left-> Right
        if start:
            traversal+= (str(start.value) + '-')
            traversal=self.PreOrderPrint(start.left, traversal)
            traversal=self.PreOrderPrint(start.right, traversal)
        return traversal

#    1
#   2  3
#  4 5  6 7
#PreOrder 1-2-7-5-3-6

tree=BinaryTree(1) #rootnode
tree.root.left=Node(2) #left child
tree.root.right=Node(3) #right child
tree.root.left.left=Node(4) #left left child
tree.root.left.right=Node(5)
```

```
tree.root.right.left=Node(6)
```

```
tree.root.left.left=Node(7)
```

```
print(tree.printTree('preorder'))
```

The screenshot shows a Python IDE with a file named 'untitled0.py'. The code defines a binary tree structure and prints its preorder traversal. The tree structure is as follows:

```
graph TD
    1((1)) --> 2((2))
    1 --> 3((3))
    2 --> 4((4))
    2 --> 5((5))
    3 --> 6((6))
    3 --> 7((7))
```

The code in the IDE is:

```
3 Created on Tue Dec 22 11:22:13 2020
4
5 @author: guneet
6 """
7 class Node(object):
8     def __init__(self, value):
9         self.left=None
10        self.right=None
11        self.value=value
12
13 class BinaryTree(object):
14     def __init__(self, root):
15         self.root=Node(root)
16
17     def printTree(self, traversalType):
18         if traversalType=='preorder':
19             return self.PreOrderPrint(self.root, '')
20
21     def PreOrderPrint(self, start, traversal):
22         #Root-> Left-> Right
23         if start:
24             traversal+= (str(start.value) + '-')
25             traversal=self.PreOrderPrint(start.left, traversal)
26             traversal=self.PreOrderPrint(start.right, traversal)
27         return traversal
28
29 #      1
30 #    2  3
31 #  4 5 6 7
32 #PreOrder 1-2-7-5-3-6
33 tree=BinaryTree(1) #rootnode
34 tree.root.left=Node(2) #left child
35 tree.root.right=Node(3) #right child
36 tree.root.left.left=Node(4) #left left child
37 tree.root.left.right=Node(5)
38 tree.root.right.left=Node(6)
39 tree.root.left.left=Node(7)
40
41 print(tree.printTree('preorder'))
```

The right sidebar shows an 'ArithmeticError' exception with the definition: `ArithmeticError(\args: object)`. The console shows the following output:

```
In [12]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py', wdir='C:/Users/gunee/Desktop/5th sem/ai')
In [13]: runfile('C:/Users/gunee/Desktop/5th sem/ai/untitled0.py', wdir='C:/Users/gunee/Desktop/5th sem/ai')
1-2-7-5-3-6-
In [14]:
```

14. WAP to implement Water Jug Problem using DFS

#Write a program to implement DEPTH First search for water jug problem

capacity = (12,8,5)

Maximum capacities of 3 jugs -> x,y,z

x = capacity[0]

y = capacity[1]

z = capacity[2]

to mark visited states

memory = {}

store solution path

ans = []

def get_all_states(state):

 # Let the 3 jugs be called a,b,c

 a = state[0]

 b = state[1]

 c = state[2]

 if(a==6 and b==6):

 ans.append(state)

 return True

 # if current state is already visited earlier

 if((a,b,c) in memory):

 return False

 memory[(a,b,c)] = 1


```
#empty jug a
if(a>0):
    #empty a into b
    if(a+b<=y):
        if( get_all_states((0,a+b,c)) ):
            ans.append(state)
            return True
    else:
        if( get_all_states((a-(y-b), y, c)) ):
            ans.append(state)
            return True
    #empty a into c
    if(a+c<=z):
        if( get_all_states((0,b,a+c)) ):
            ans.append(state)
            return True
    else:
        if( get_all_states((a-(z-c), b, z)) ):
            ans.append(state)
            return True

#empty jug b
if(b>0):
    #empty b into a
    if(a+b<=x):
        if( get_all_states((a+b, 0, c)) ):
            ans.append(state)
            return True
    else:
        if( get_all_states((x, b-(x-a), c)) ):
            ans.append(state)
            return True
    #empty b into c
    if(b+c<=z):
        if( get_all_states((a, 0, b+c)) ):
```

```
        ans.append(state)
        return True
    else:
        if( get_all_states((a, b-(z-c), z)) ):
            ans.append(state)
            return True

#empty jug c
if(c>0):
    #empty c into a
    if(a+c<=x):
        if( get_all_states((a+c, b, 0)) ):
            ans.append(state)
            return True
    else:
        if( get_all_states((x, b, c-(x-a))) ):
            ans.append(state)
            return True

#empty c into b
if(b+c<=y):
    if( get_all_states((a, b+c, 0)) ):
        ans.append(state)
        return True
    else:
        if( get_all_states((a, y, c-(y-b))) ):
            ans.append(state)
            return True

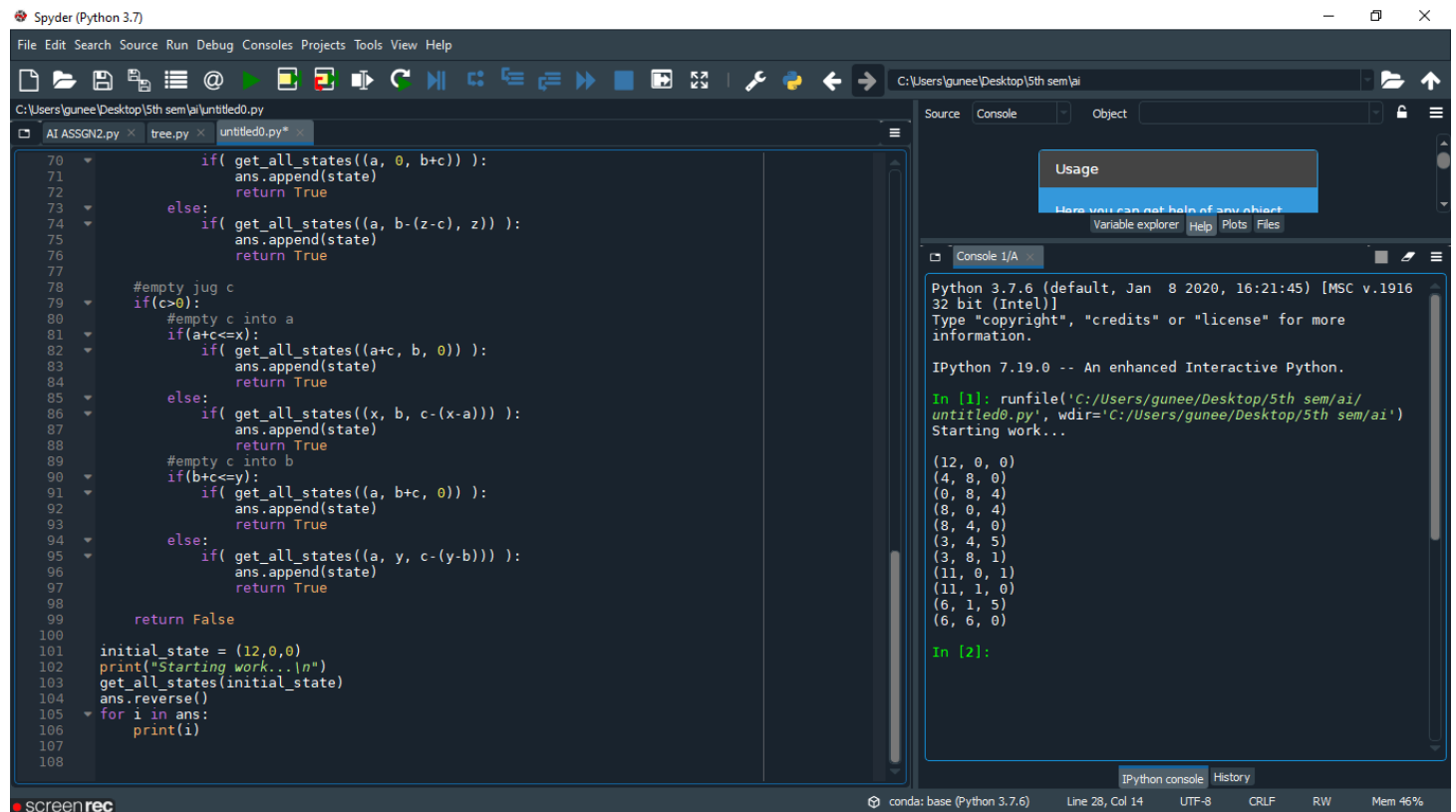
return False

initial_state = (12,0,0)
print("Starting work...\n")
get_all_states(initial_state)
```

```
ans.reverse()
```

```
for i in ans:
```

```
    print(i)
```



The screenshot shows the Spyder Python IDE with a file named 'untitled0.py' open. The code in the editor is as follows:

```
70     if( get_all_states((a, 0, b+c)) ):
71         ans.append(state)
72         return True
73     else:
74         if( get_all_states((a, b-(z-c), z)) ):
75             ans.append(state)
76             return True
77
78     #empty jug c
79     if(c>0):
80         #empty c into a
81         if(a+c<=x):
82             if( get_all_states((a+c, b, 0)) ):
83                 ans.append(state)
84                 return True
85         else:
86             if( get_all_states((x, b, c-(x-a))) ):
87                 ans.append(state)
88                 return True
89         #empty c into b
90         if(b+c<=y):
91             if( get_all_states((a, b+c, 0)) ):
92                 ans.append(state)
93                 return True
94         else:
95             if( get_all_states((a, y, c-(y-b))) ):
96                 ans.append(state)
97                 return True
98
99     return False
100
101 initial_state = (12,0,0)
102 print("Starting work...\n")
103 get_all_states(initial_state)
104 ans.reverse()
105 for i in ans:
106     print(i)
107
108
```

The console output shows the following:

```
Python 3.7.6 (default, Jan 8 2020, 16:21:45) [MSC v.1916
32 bit (Intel)]
Type "copyright", "credits" or "license" for more
information.

IPython 7.19.0 -- An enhanced Interactive Python.

In [1]: runfile('C:/Users/gunee/Desktop/5th sem/ai/
untitled0.py', wdir='C:/Users/gunee/Desktop/5th sem/ai')
Starting work...

(12, 0, 0)
(4, 8, 0)
(0, 8, 4)
(8, 0, 4)
(8, 4, 0)
(3, 4, 5)
(3, 8, 1)
(11, 0, 1)
(11, 1, 0)
(6, 1, 5)
(6, 6, 0)

In [2]:
```