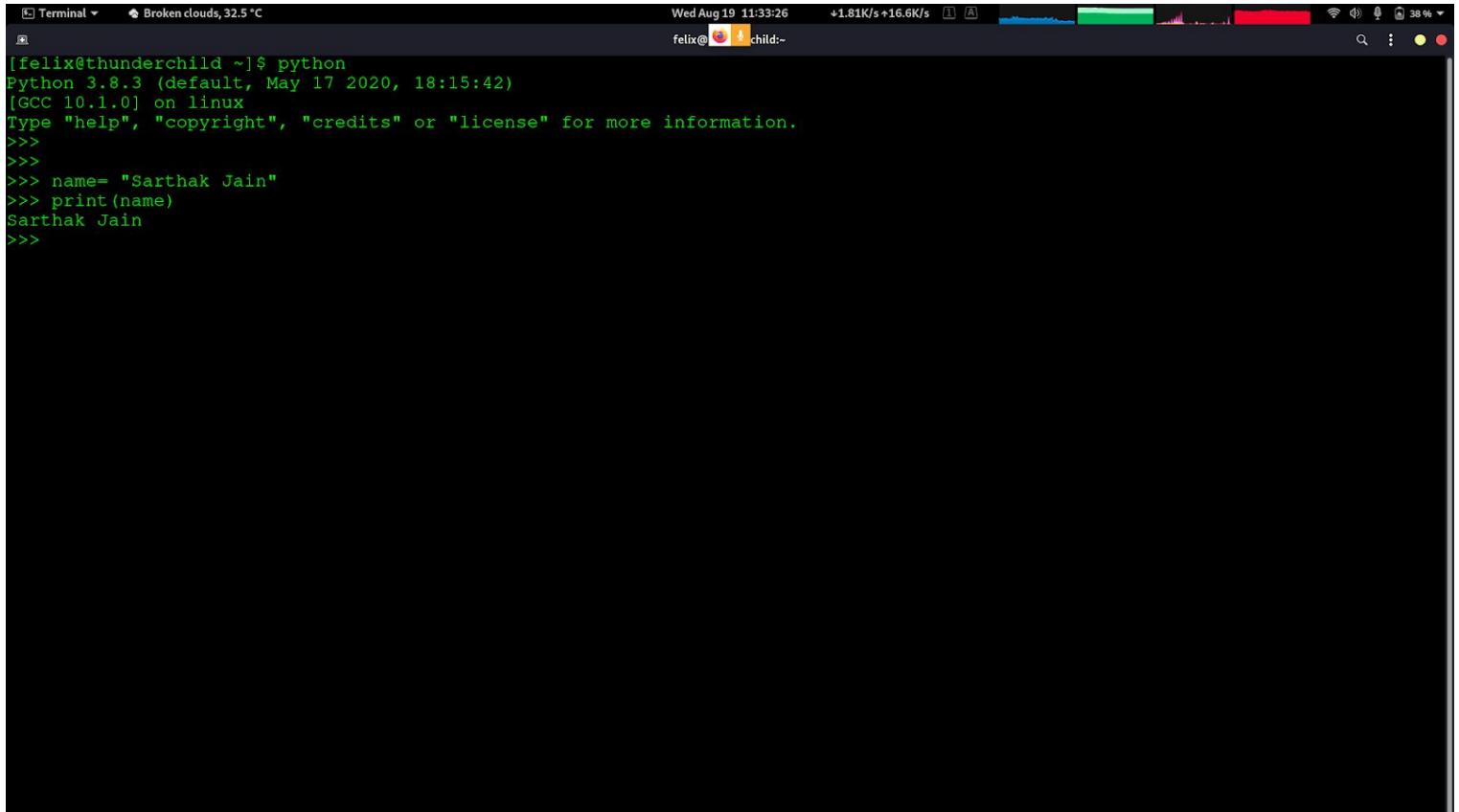


Name: Sarthak Jain  
Programme: B.tech CSE AI&ML (Batch 5)  
Course: Data Mining and Prediction by Machines

---

## Lab 1 (Python Basics)

### 1. Python Program to Print name

A screenshot of a terminal window. The window title bar shows 'Terminal', 'Broken clouds, 32.5 °C', 'Wed Aug 19 11:33:26', '+1.81K/s +16.6K/s', and system icons. The terminal content shows a user running 'python' in a shell. The output shows 'Python 3.8.3 (default, May 17 2020, 18:15:42)', '[GCC 10.1.0] on linux', and a prompt to type 'help', 'copyright', 'credits' or 'license' for more information. The user then enters a multi-line Python program: '>>>' followed by 'name= "Sarthak Jain"', 'print(name)', and another '>>>'. The output of the program is 'Sarthak Jain'.

### 2. Python Program to Add Two Numbers

```
>>> a= 32
>>> b= 43
>>>
>>> print("a+b is", a+b)
a+b is 75
```

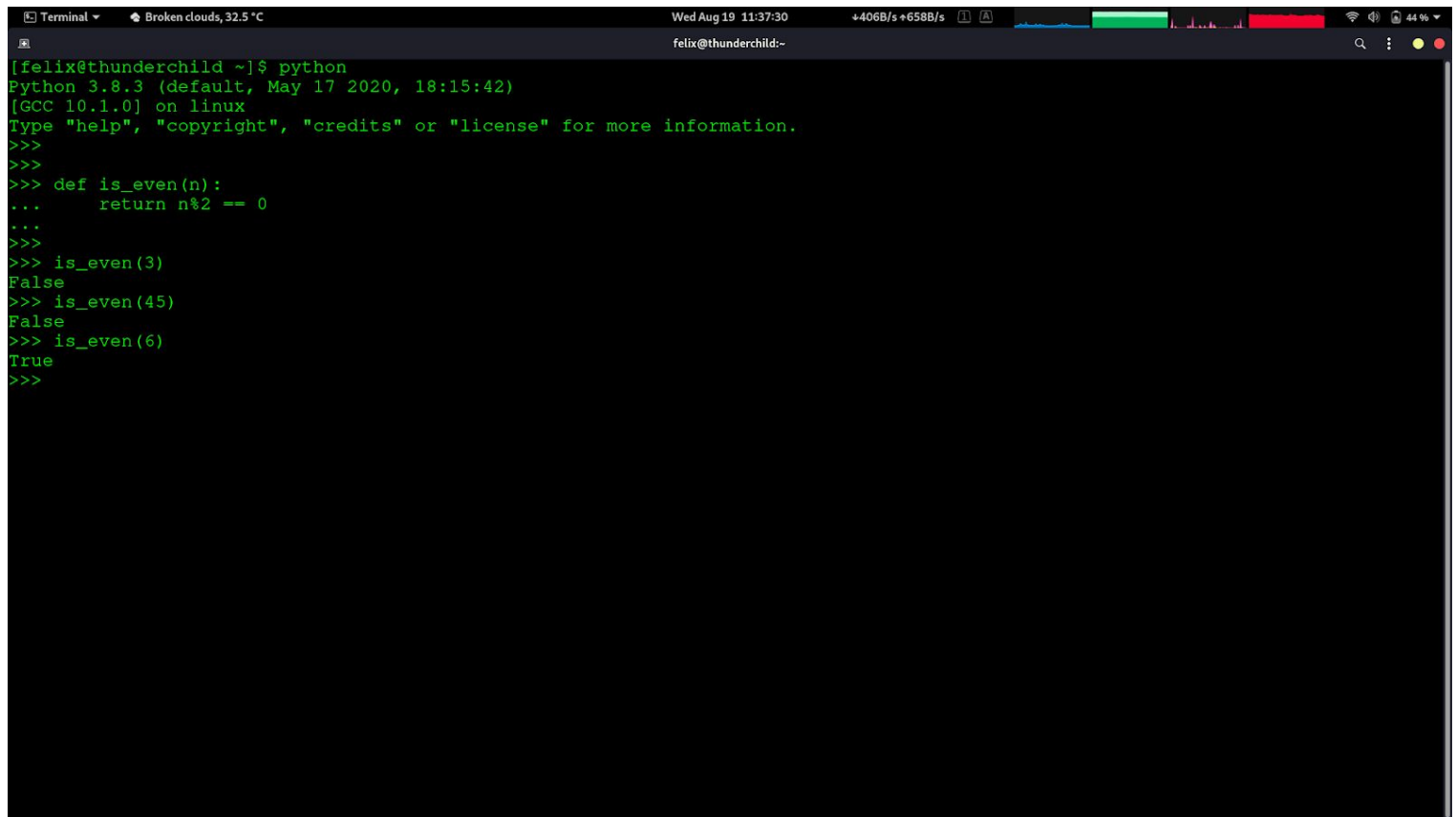
```
Terminal Broken clouds, 32.5 °C Wed Aug 19 11:35:03 +281B/s +627B/s felix@thunderchild:~  
[felix@thunderchild ~]$ python  
Python 3.8.3 (default, May 17 2020, 18:15:42)  
[GCC 10.1.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
>>> a= 32  
>>> b= 43  
>>>  
>>> print("a+b is", a+b)  
a+b is 75  
>>>
```

### 3. Python Program to Swap Two Variables

```
Terminal Broken clouds, 32.5 °C Wed Aug 19 11:36:27 +1.04K/s +1.45K/s felix@thunderchild:~  
[felix@thunderchild ~]$ python  
Python 3.8.3 (default, May 17 2020, 18:15:42)  
[GCC 10.1.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
>>>  
>>> a= 45564  
>>> b= 2323  
>>>  
>>> a, b= b, a  
>>>  
>>> print(a, b)  
2323 45564  
>>> █
```

#### 4. Python Program to Check if a Number is Odd or Even

```
>>> def is_even(n):
...     return n%2 == 0
...
>>>
>>> is_even(3)
False
>>> is_even(45)
False
>>> is_even(6)
True
>>>
```

A screenshot of a terminal window with a dark background. The window title bar shows 'Terminal', 'Broken clouds, 32.5 °C', 'Wed Aug 19 11:37:30', '+406B/s +658B/s', and system icons for network, battery, and 44% battery level. The terminal content shows a user running 'python' in a shell. The output shows 'Python 3.8.3 (default, May 17 2020, 18:15:42)' and '[GCC 10.1.0] on linux'. The user then enters a Python script to define a function 'is\_even(n)' that returns 'n%2 == 0'. The script is executed, and the results for 'is\_even(3)', 'is\_even(45)', and 'is\_even(6)' are shown as 'False', 'False', and 'True' respectively.

```
[felix@thunderchild ~]$ python
Python 3.8.3 (default, May 17 2020, 18:15:42)
[GCC 10.1.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
>>>
>>> def is_even(n):
...     return n%2 == 0
...
>>>
>>> is_even(3)
False
>>> is_even(45)
False
>>> is_even(6)
True
>>>
```

#### 5. Python Program to Check Prime Number

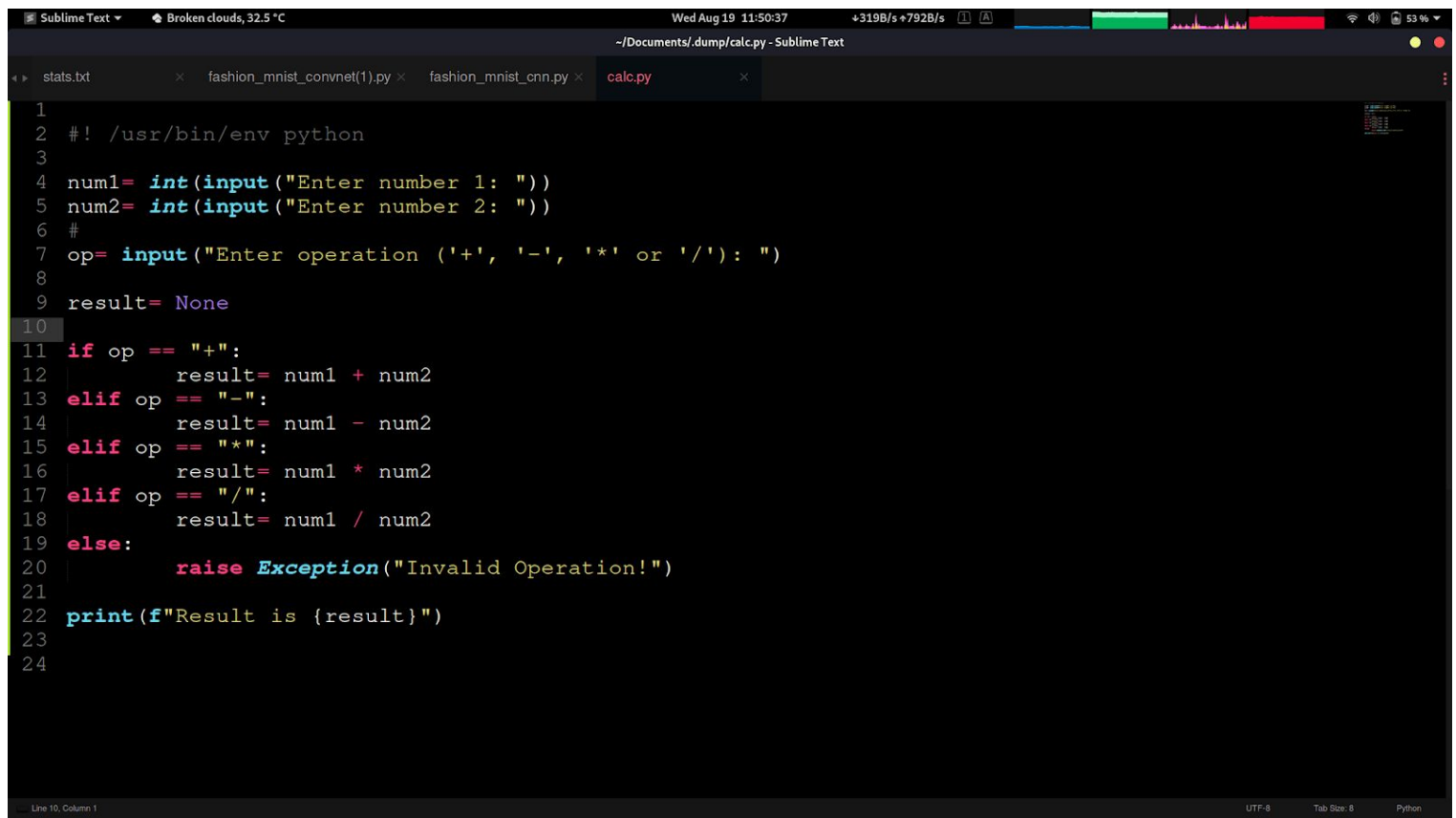
```
>>> def check_prime(n):
...     for i in range(2, int(n**0.5)+1):
...         if n%i == 0:
...             return False
...     return True
...
>>>
>>> check_prime(3)
True
>>> check_prime(32)
False
```

```
Terminal Broken clouds, 32.5 °C Wed Aug 19 11:41:44 +273B/s +659B/s felix@thunderchild:~  
[felix@thunderchild ~]$ python  
Python 3.8.3 (default, May 17 2020, 18:15:42)  
[GCC 10.1.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
>>>  
>>> def check_prime(n):  
...     for i in range(2, int(n**0.5)+1):  
...         if n%i == 0:  
...             return False  
...     return True  
...  
>>>  
>>> check_prime(3)  
True  
>>> check_prime(32)  
False  
>>>
```

## 6. Python Program to Display the multiplication Table

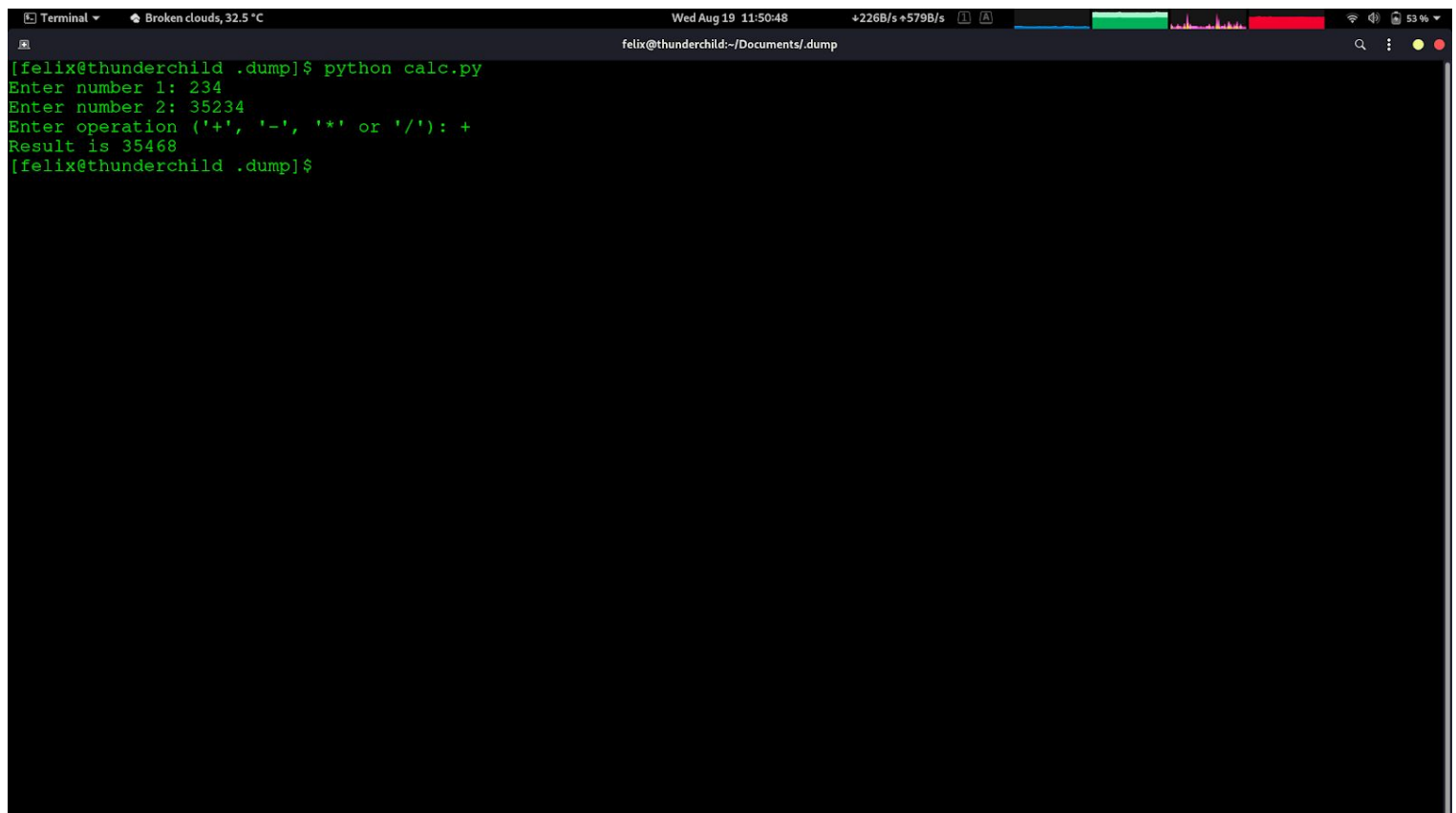
```
Terminal Broken clouds, 32.5 °C Wed Aug 19 11:43:14 +378B/s +717B/s felix@thunderchild:~  
[felix@thunderchild ~]$ python  
Python 3.8.3 (default, May 17 2020, 18:15:42)  
[GCC 10.1.0] on linux  
Type "help", "copyright", "credits" or "license" for more information.  
>>>  
>>> def display_table(n):  
...     for i in range(1, 11):  
...         print(f"{n} * {i} = {n*i}")  
...  
>>>  
>>> display_table(14)  
14 * 1 = 14  
14 * 2 = 28  
14 * 3 = 42  
14 * 4 = 56  
14 * 5 = 70  
14 * 6 = 84  
14 * 7 = 98  
14 * 8 = 112  
14 * 9 = 126  
14 * 10 = 140  
>>>
```

## 7. Python Program to Make a Simple Calculator



The screenshot shows the Sublime Text editor with a file named `calc.py` open. The code is a Python script for a simple calculator. It prompts the user to enter two numbers and an operation. The operations supported are addition (+), subtraction (-), multiplication (\*), and division (/). If an invalid operation is entered, it raises an exception. The result is then printed.

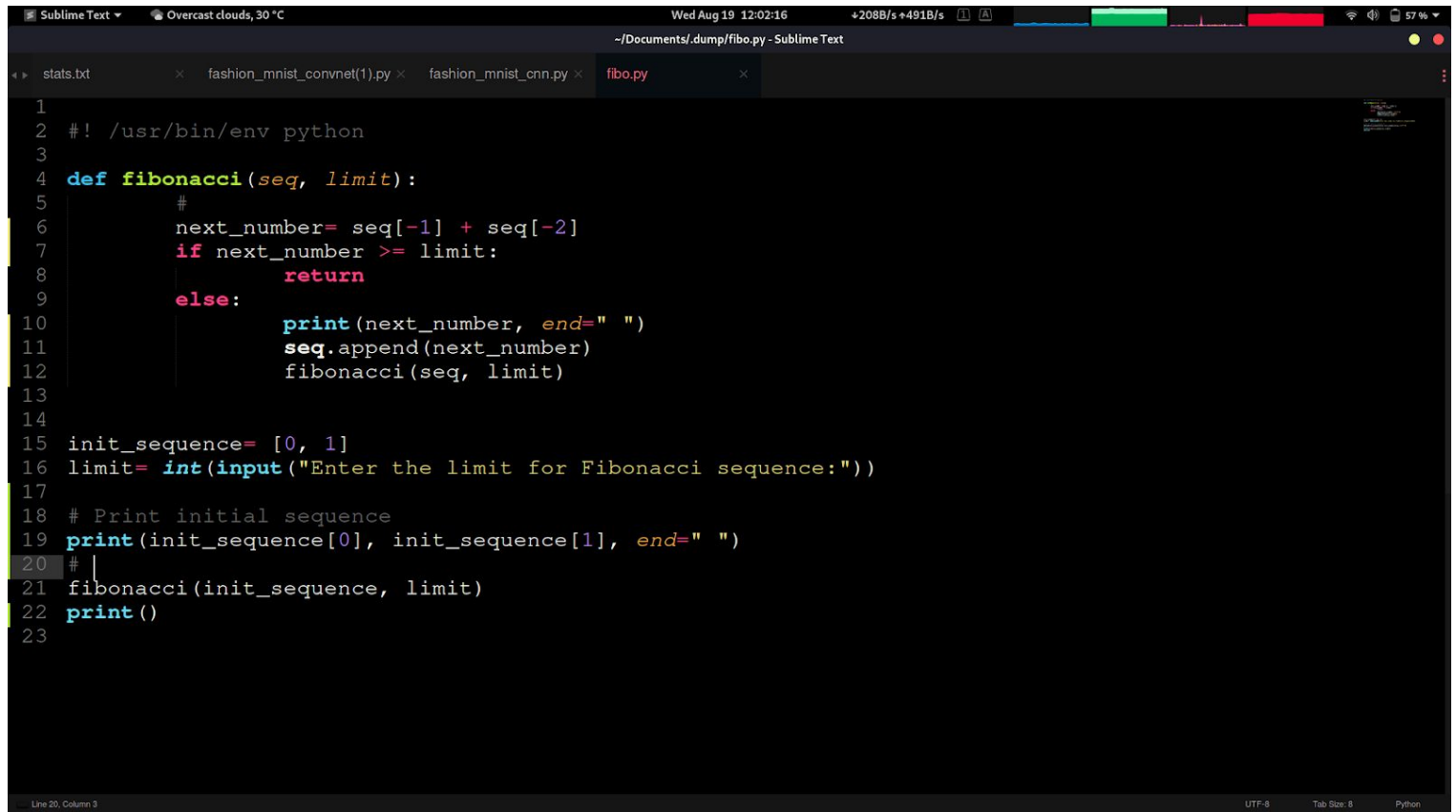
```
1
2 #!/usr/bin/env python
3
4 num1= int(input("Enter number 1: "))
5 num2= int(input("Enter number 2: "))
6 #
7 op= input("Enter operation ('+', '-', '*' or '/'): ")
8
9 result= None
10
11 if op == "+":
12     result= num1 + num2
13 elif op == "-":
14     result= num1 - num2
15 elif op == "*":
16     result= num1 * num2
17 elif op == "/":
18     result= num1 / num2
19 else:
20     raise Exception("Invalid Operation!")
21
22 print(f"Result is {result}")
23
24
```



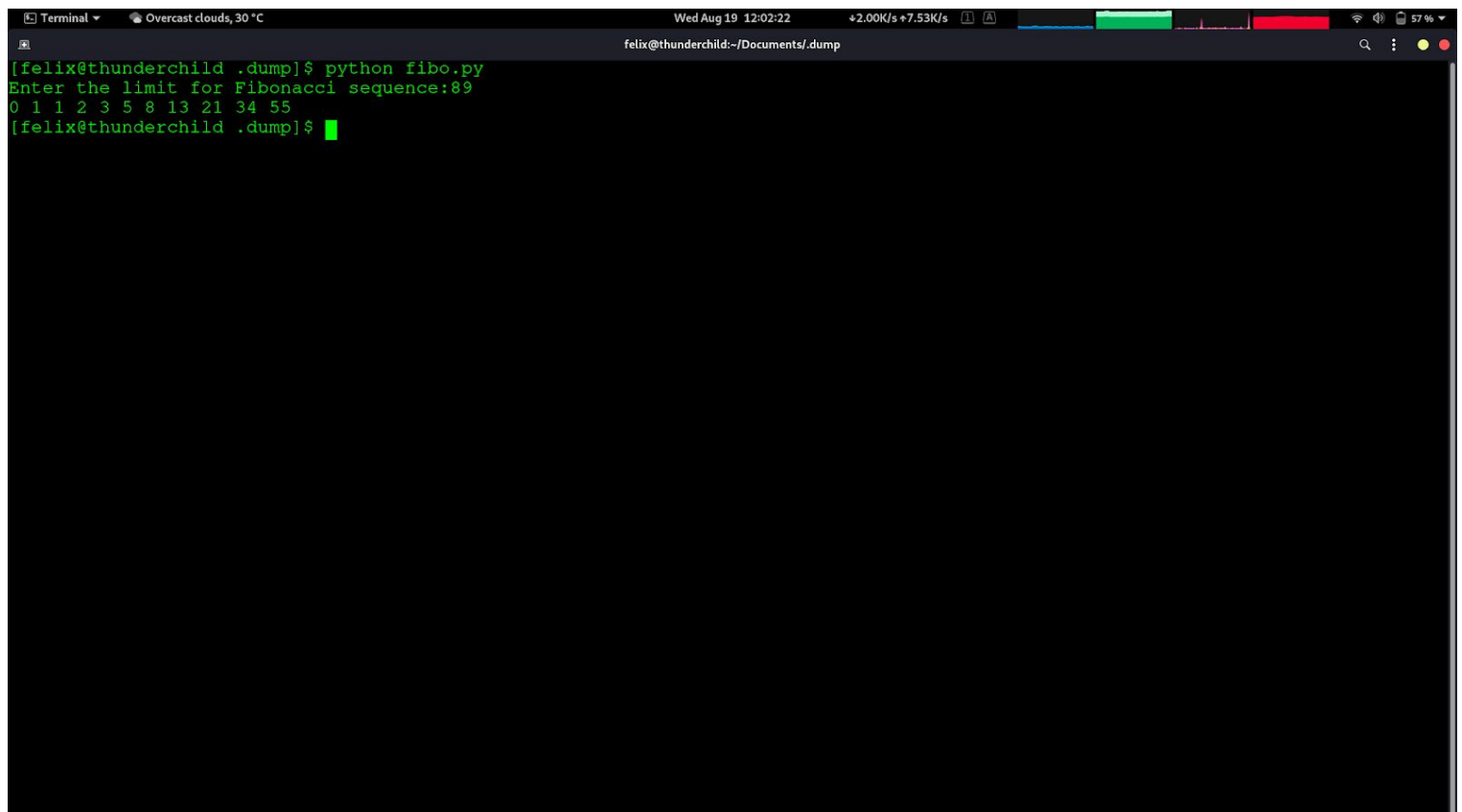
The screenshot shows a terminal window where the `calc.py` program has been executed. The user has entered the numbers 234 and 35234, and the operation '+'. The program has calculated the result as 35468.

```
[felix@thunderchild .dump]$ python calc.py
Enter number 1: 234
Enter number 2: 35234
Enter operation ('+', '-', '*' or '/'): +
Result is 35468
[felix@thunderchild .dump]$
```

## 8. Python Program to Display Fibonacci Sequence Using Recursion



```
1
2 #!/usr/bin/env python
3
4 def fibonacci(seq, limit):
5     #
6     next_number= seq[-1] + seq[-2]
7     if next_number >= limit:
8         return
9     else:
10        print(next_number, end=" ")
11        seq.append(next_number)
12        fibonacci(seq, limit)
13
14
15 init_sequence= [0, 1]
16 limit= int(input("Enter the limit for Fibonacci sequence:"))
17
18 # Print initial sequence
19 print(init_sequence[0], init_sequence[1], end=" ")
20 # |
21 fibonacci(init_sequence, limit)
22 print()
23
```



```
[felix@thunderchild .dump]$ python fibo.py
Enter the limit for Fibonacci sequence:89
0 1 1 2 3 5 8 13 21 34 55
[felix@thunderchild .dump]$
```

## 9. Python Program to Add Two Matrices

```
Sublime Text  Overcast clouds, 30 °C  Wed Aug 19 12:13:45  +207B/s +630B/s  58 %
~/Documents/dump/add_matrices.py - Sublime Text

stats.txt  fashion_mnist_convnet(1).py  fashion_mnist_cnn.py  fibo.py  add_matrices.py

1
2  #!/usr/bin/env python
3
4  def add_matrices(a, b, dim):
5      c = [*b]
6      for y in range(dim[0]):
7          for x in range(dim[1]):
8              c[y][x] = a[y][x] + b[y][x]
9      #
10     return c
11
12  def print_matrix(a, dim):
13      for y in range(dim[0]):
14          for x in range(dim[1]):
15              print(a[y][x], end="\t")
16          print()
17      print()
18
19  def input_matrix(dim):
20      a = []
21      for y in range(dim[0]):
22          a.append([])
23          for x in range(dim[1]):
24              n = int(input(f"Enter number at {y},{x}:"))
25              a[y].append(n)
26      #
27      return a
28
29  const_dim = (3, 3)
```

```
Sublime Text  Overcast clouds, 30 °C  Wed Aug 19 12:13:50  +486B/s +942B/s  58 %
~/Documents/dump/add_matrices.py - Sublime Text

stats.txt  fashion_mnist_convnet(1).py  fashion_mnist_cnn.py  fibo.py  add_matrices.py

16      print()
17      print()
18
19  def input_matrix(dim):
20      a = []
21      for y in range(dim[0]):
22          a.append([])
23          for x in range(dim[1]):
24              n = int(input(f"Enter number at {y},{x}:"))
25              a[y].append(n)
26      #
27      return a
28
29  const_dim = (3, 3)
30  print("Enter 3x3 Matrix A:")
31  a = input_matrix(const_dim)
32  print("Matrix A is ")
33  print_matrix(a, const_dim)
34
35  print("Enter 3x3 Matrix B:")
36  b = input_matrix(const_dim)
37  print("Matrix B is ")
38  print_matrix(b, const_dim)
39
40  c = add_matrices(a, b, const_dim)
41  print("A+B is ")
42  print_matrix(c, const_dim)
43
```

```
Terminal Overcast clouds, 30 °C Wed Aug 19 12:14:12 +7.49K/s +73.5K/s felix@thunderchild:~/Documents/.dump
[felix@thunderchild .dump]$ python add_matrices.py
Enter 3x3 Matrix A:
Enter number at 0,0:324
Enter number at 0,1:34
Enter number at 0,2:23
Enter number at 1,0:4
Enter number at 1,1:23
Enter number at 1,2:23
Enter number at 2,0:23
Enter number at 2,1:2
Enter number at 2,2:32
Matrix A is
324    34    23
4      23    23
23     2     32

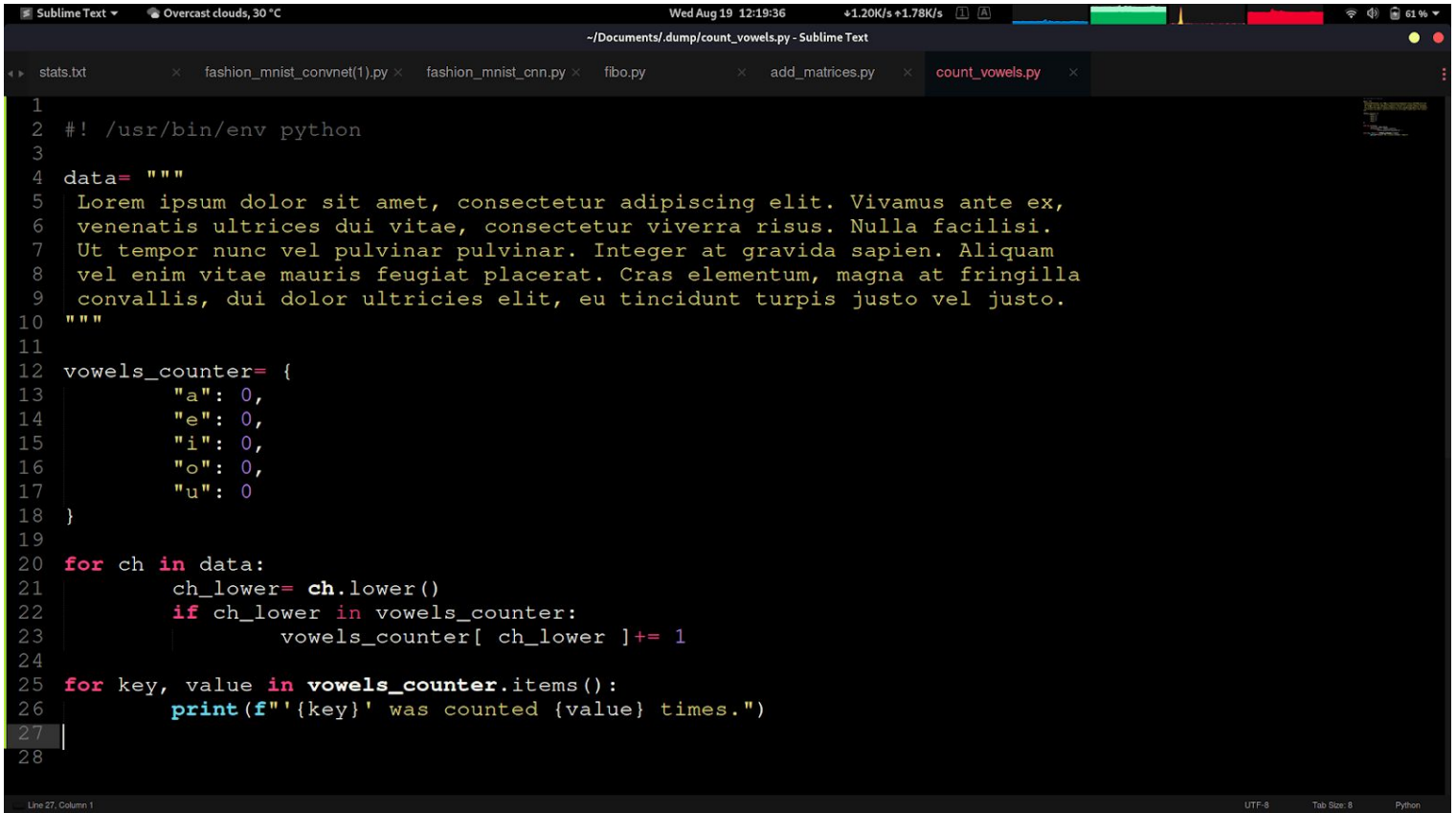
Enter 3x3 Matrix B:
Enter number at 0,0:23
Enter number at 0,1:42
Enter number at 0,2:4
Enter number at 1,0:23
Enter number at 1,1:423
Enter number at 1,2:42
Enter number at 2,0:34
Enter number at 2,1:34
Enter number at 2,2:234
Matrix B is
23     42     4
23     423    42
34     34    234

A+B is
347    76     27
27     446    65
57     36    266

[felix@thunderchild .dump]$
```

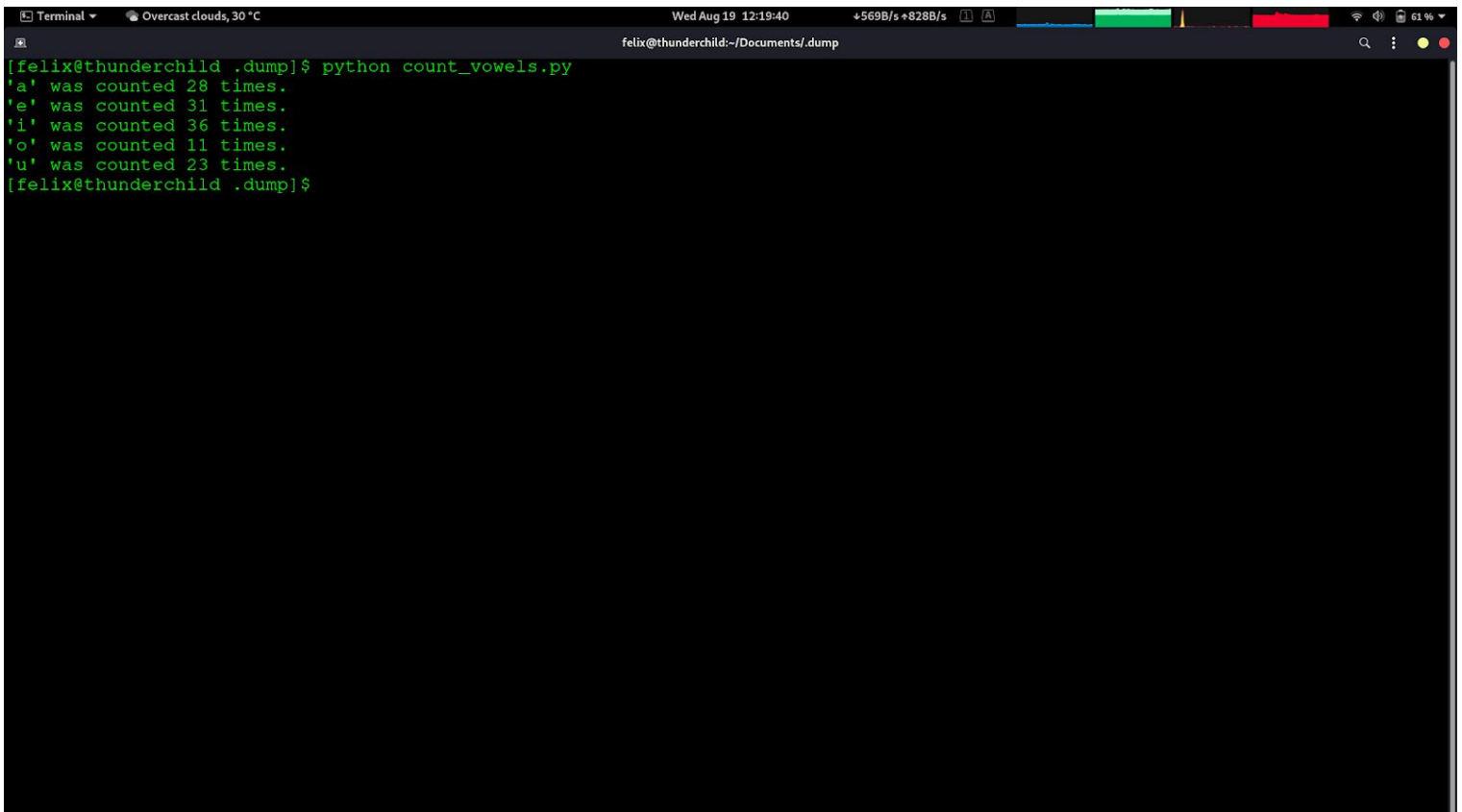


## 10. Python Program to Count the Number of Each Vowel



The screenshot shows a Sublime Text editor window with the file path `~/Documents/.dump/count_vowels.py`. The script defines a string `data` containing Lorem Ipsum text and a dictionary `vowels_counter` to track the frequency of vowels. It then iterates through the text, counts each vowel, and prints the results.

```
1
2 #!/usr/bin/env python
3
4 data= """
5 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Vivamus ante ex,
6 venenatis ultrices dui vitae, consectetur viverra risus. Nulla facilisi.
7 Ut tempor nunc vel pulvinar pulvinar. Integer at gravida sapien. Aliquam
8 vel enim vitae mauris feugiat placerat. Cras elementum, magna at fringilla
9 convallis, dui dolor ultricies elit, eu tincidunt turpis justo vel justo.
10 """
11
12 vowels_counter= {
13     "a": 0,
14     "e": 0,
15     "i": 0,
16     "o": 0,
17     "u": 0
18 }
19
20 for ch in data:
21     ch_lower= ch.lower()
22     if ch_lower in vowels_counter:
23         vowels_counter[ ch_lower ]+= 1
24
25 for key, value in vowels_counter.items():
26     print(f"'{key}' was counted {value} times.")
27
28
```



The screenshot shows a terminal window with the command `python count_vowels.py` executed. The output displays the count for each vowel: 'a' (28), 'e' (31), 'i' (36), 'o' (11), and 'u' (23).

```
[felix@thunderchild .dump]$ python count_vowels.py
'a' was counted 28 times.
'e' was counted 31 times.
'i' was counted 36 times.
'o' was counted 11 times.
'u' was counted 23 times.
[felix@thunderchild .dump]$
```