

Computer **Science**

Class XII ***Practical Journal***

Submitted By:-

ADITYA SAMBHAJI BABAR
(XII - SCIENCE)

**Shri Vasantao Banduji Patil Trust's
APPASAHEB BIRNALE PUBLIC SCHOOL, SANGLI
SENIOR SECONDRY**

(AFFILIATED TO CBSE, NEW DELHI, NO: 1130074)

Behind Railway Station, Shinde Mala, Sangli., Maharashta



DEPARTMENT OF COMPUTER SCIENCE

CERTIFICATE

**Certified that this is a bonafide record of the project work in
COMPUTER SCIENCE**

**Submitted to the Central Board of Secondary Education in
partial fulfillment of**

Senior Secondary Examination.

DATE

Principal

Submitted for Practicle Examination and VivaVoice held on _____

Internal Examiner

External Examiner

DECLARATION

I, ADITYA SAMBHAJI BABAR, hereby declare that this work entitled "Class XII Practical Journal" submitted to APPASAHEB BIRNALE PUBLIC SCHOOL

SENIOR SECONDRY, Shinde Mala, Sangli., Maharashtra, 416416 (Affiliated to CBSE, New Delhi, and Affiliation No. 1130074) is original record work done by me under the supervision and guidance of Mr. Sushant Pawar, Department of Computer Science.

Name of Candidate

Register No.

**Signature of the
Candidate**

Countersigned By:

Mr. Sushant Pawar
Teacher incharge
Department of Computer Science
Appasaheb Birnale Public School
Senior Secondary, Shinde Mala
Place: Sangli

Date : _____

ACKNOWLEDGEMENT

The success and final outcome of this project required a lot of guidance and assistance from many people and I am extremely fortunate to have got this all along the completions of my project work. Whatever I have done is only due to such guidance and assistance and i would not forget to thank them.

I have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals. I would like to extend my sincere thanks to all of them.

I am highly indepted to my computer teacher Mr. Sushant Pawar for his guidance and constant supervision as well as for providing necessary information regarding the project & also for supporting me in completing the project. I also thank our respected Principal Shradha khot for their support.

I am thankful to and fortunate enough to get constent encouragment, support and guidance from all teaching staff of Department of computer science which helped us in succesfully completing this project.

I would like to express my gratitude towards my parents for their kinf co-operation and encouragment which help me in completion of this project.

My thanks and appreciations also go to my colleague in developing the project and people who have willingly helped me out with their abilities.

INDEX-PYTHON

S. NO.	Content
1.	Program to generate a pattern
2.	Program to arrange dictionary in accending order
3.	Working with functions to calculate sum of squares
4.	Program to find the area of a regular polygon
5.	File handling : Edit a phone number
6.	Data visuallisation using pyplot-bar chart
7.	Data visuallisation using pyplot-line chart
8.	Data visuallisation using pyplot-pie chart
9.	Data stucture : to insert an element in a sorted list
10.	Data structure : to insert in a reverse order
11.	Binary Search
12.	Fibonacci Series
13.	Factoral Number
14.	File Handling-Absolute Path, Reverse Path
15.	File Handling
16.	SQL connection

Index-MySQL

**Sr.
No.**

Program Name

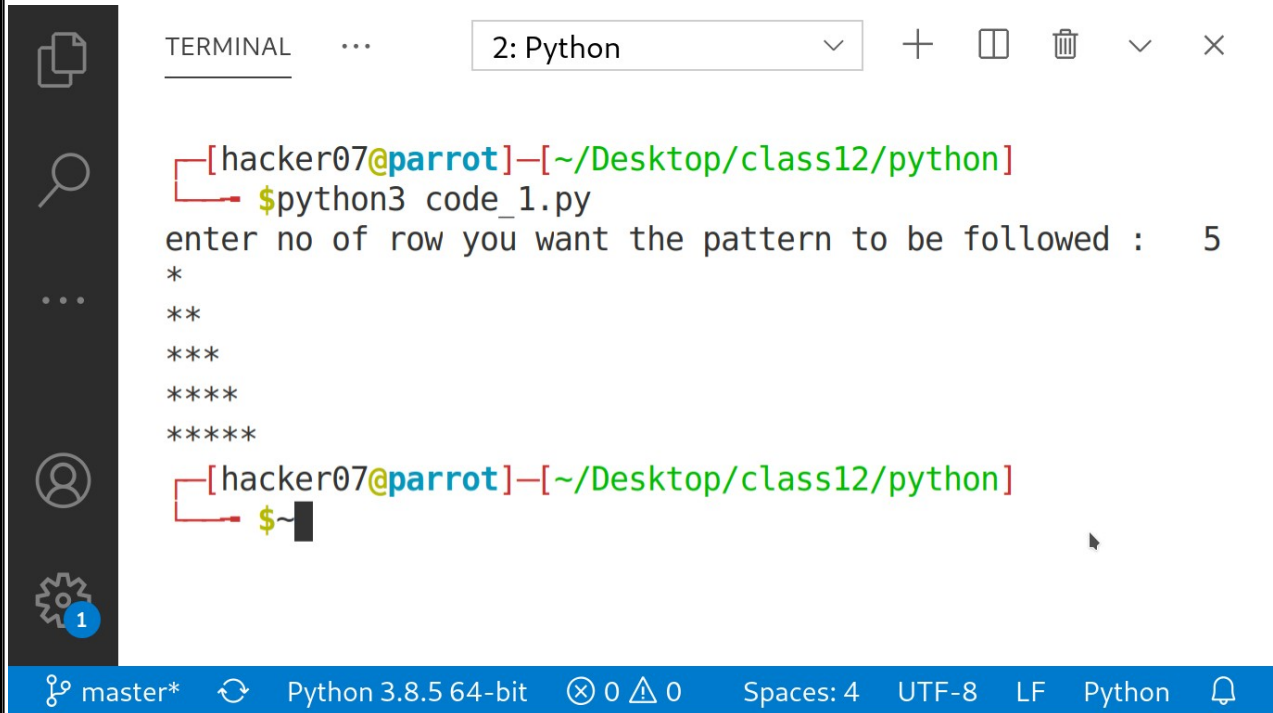
- | | |
|----|------------------------------------|
| 1. | Create Table, Entering Data |
| 2. | Joining Table |
| 3. | Performing Arithmetic operations |
| 4. | Grouping by using groups by clause |
| 5. | Arranging by using Order by clause |

01. Write a program to generate patterns

```
no_of_rows = int(input("enter no of row you want the pattern to be followed :\t"))
```

```
for i in range(no_of_rows):  
    for _ in range(i+1):  
        print("*", end="")  
    print()
```

Output :-



The image shows a terminal window with a dark theme. The title bar at the top reads "TERMINAL" followed by a dropdown menu showing "2: Python". The terminal content shows a user prompt "[hacker07@parrot]~" and a command "\$python3 code_1.py". The output of the script is a pattern of asterisks: a single asterisk on the first line, followed by two, three, four, and five asterisks on the subsequent lines. The user has entered "5" in response to the prompt "enter no of row you want the pattern to be followed :". The terminal status bar at the bottom indicates the current directory is "master*", the Python version is "Python 3.8.5 64-bit", and the encoding is "UTF-8".

```
TERMINAL  ...  2: Python  +  [icon]  [icon]  [icon]  [icon]  [icon]

[hacker07@parrot]~[~/Desktop/class12/python]
$python3 code_1.py
enter no of row you want the pattern to be followed : 5
*
**
***
****
*****
[hacker07@parrot]~[~/Desktop/class12/python]
$~
```

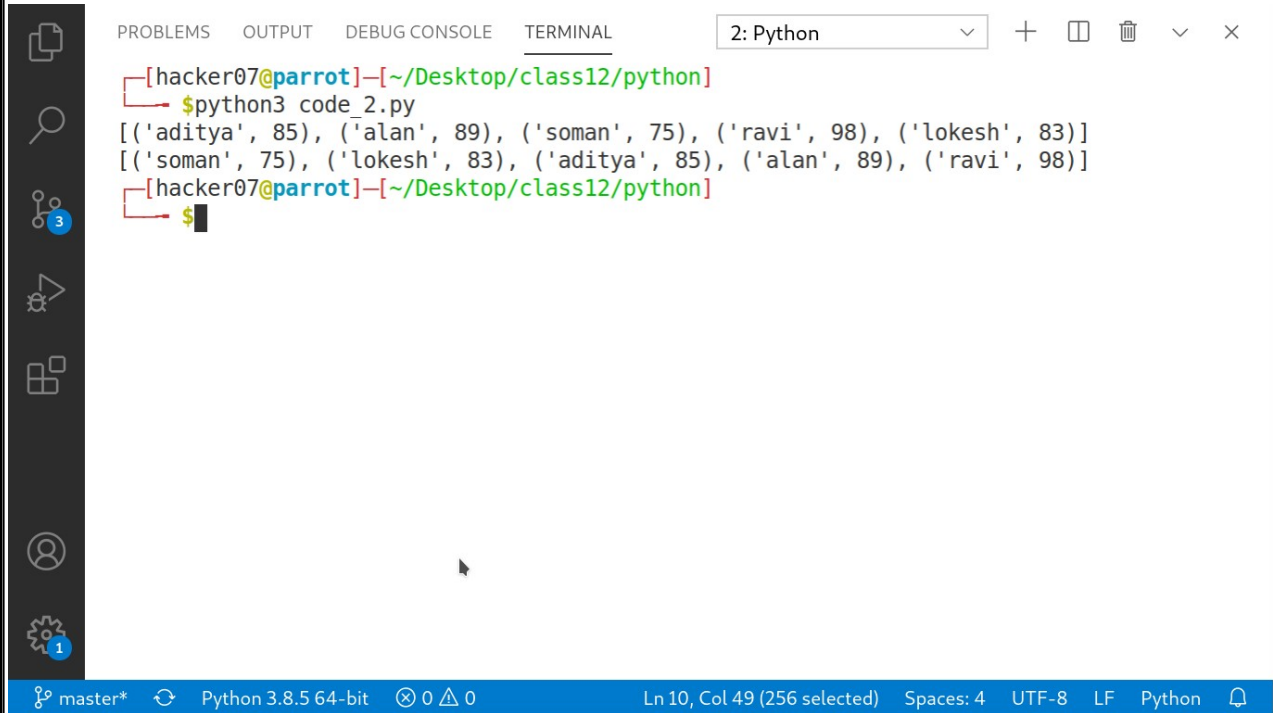
master* Python 3.8.5 64-bit 0 0 Spaces: 4 UTF-8 LF Python [icon]

02. Program to arrange dictionary in accending order

```
data = {"aditya":85, "alan":89, "soman":75, "ravi":98,  
"lokesh":83}
```

```
a = []  
for key, values in data.items():  
    b = (key, values)  
    a.append(b)  
print(a)  
print(sorted(a, key= lambda kv: (kv[1], kv[0])))
```

Output:-



The screenshot shows a VS Code terminal window with the following content:

```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_2.py
[('aditya', 85), ('alan', 89), ('soman', 75), ('ravi', 98), ('lokes', 83)]
[('soman', 75), ('lokes', 83), ('aditya', 85), ('alan', 89), ('ravi', 98)]
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

The terminal window has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Settings. The top bar shows the terminal title '2: Python' and various window controls. The bottom status bar indicates the current file is 'master*', the Python version is 'Python 3.8.5 64-bit', and the cursor is at 'Ln 10, Col 49 (256 selected)'.

03. Write a program that receives 4 digit no. and calculate sum of squares of first 2 digits and last 2 digits.

```
import math
```

```
def sum_of_squares(number):
```

```
    x = int(number[0:2])
```

```
    y = int(number[2:5])
```

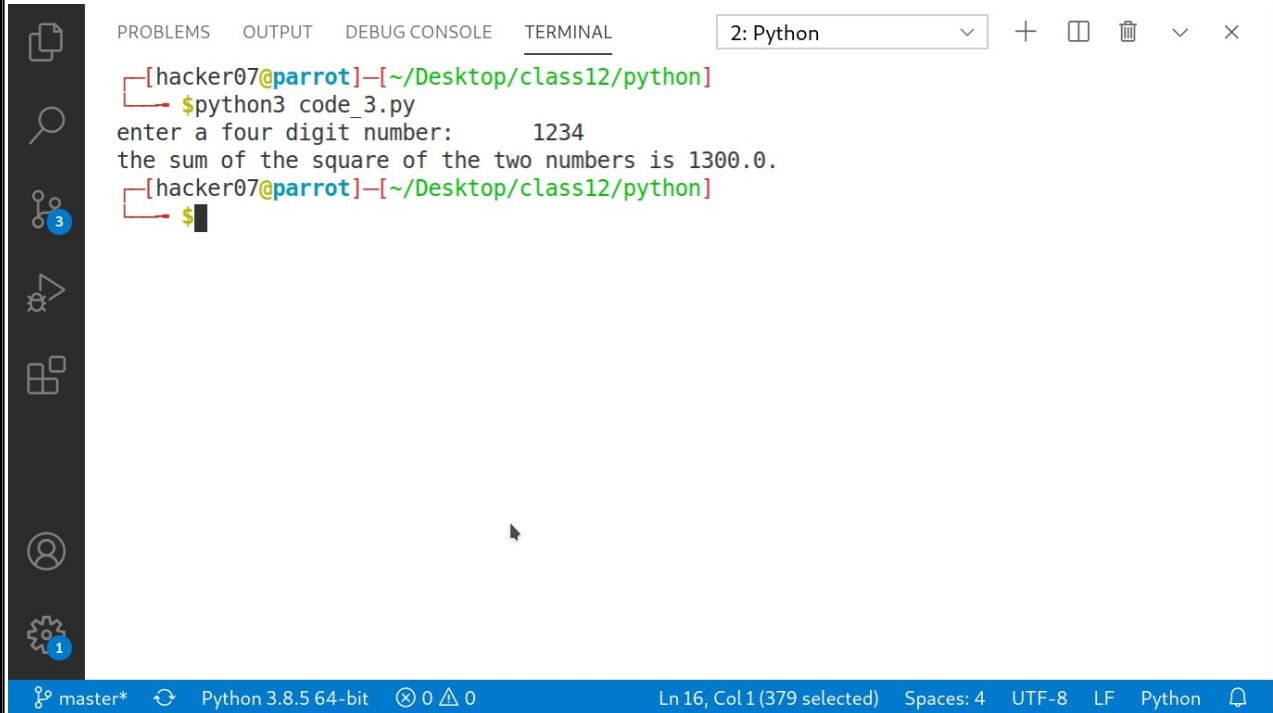
```
    z = math.pow(x, 2) + math.pow(y, 2)
```

```
    print(f"the sum of the square of the two numbers is {z}.")
```

```
number = input("enter a four digit number:\t")
```

```
sum_of_squares(number)
```

Output:-



The screenshot shows a VS Code interface with a terminal window open. The terminal title bar indicates it is a Python file named '2: Python'. The terminal content shows a user prompt, a command to run a Python script, the script's output, and a subsequent shell prompt.

```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_3.py
enter a four digit number:      1234
the sum of the square of the two numbers is 1300.0.
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

The status bar at the bottom of the editor shows the following information: 'master*' (branch), 'Python 3.8.5 64-bit' (interpreter), '0 0' (diff status), 'Ln 16, Col 1 (379 selected)' (cursor position), 'Spaces: 4' (indentation), 'UTF-8' (encoding), 'LF' (line endings), 'Python' (language), and a bell icon for notifications.

04. write a program to find the area of a regular polygon.

```
from math import tan, pi
```

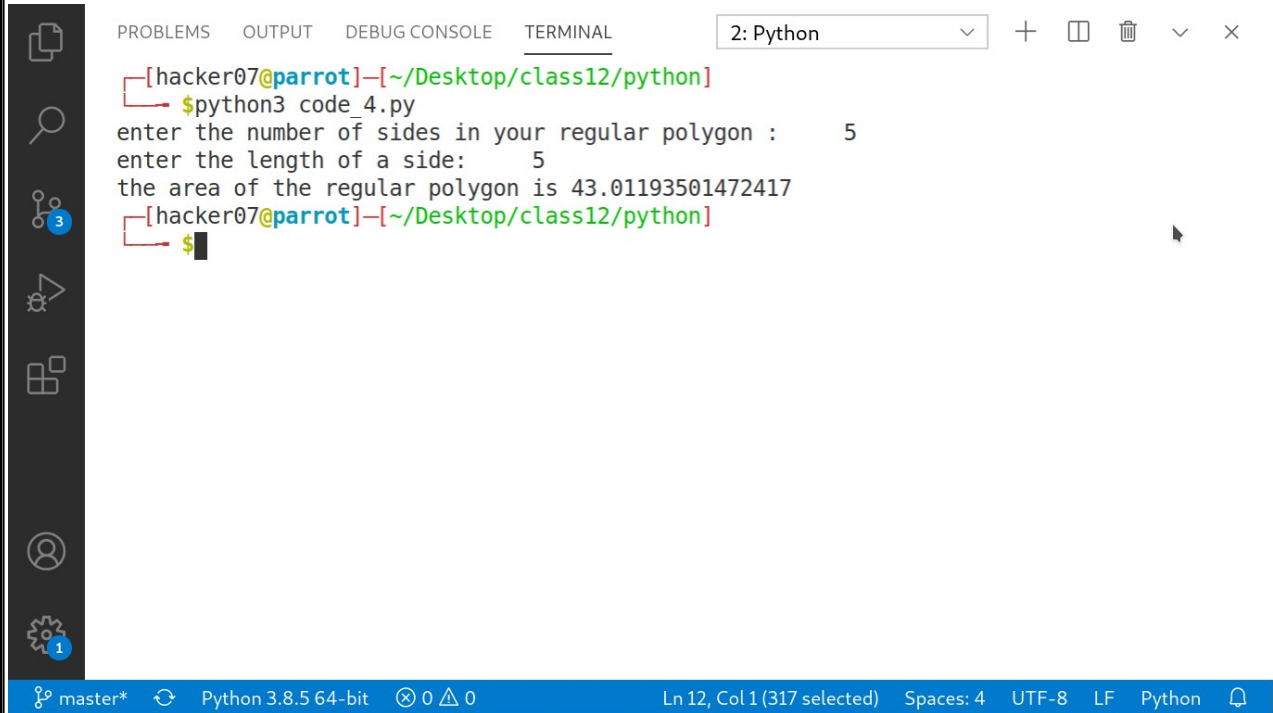
```
no_sides = int(input("enter the number of sides in your regular polygon : \t"))
```

```
length = int(input("enter the length of a side: \t"))
```

```
a = (length**2)*no_sides/(4 * tan(pi/no_sides))
```

```
print(f"the area of the regular polygon is {a} ")
```

Output:-



The screenshot shows a VS Code interface with a terminal window open. The terminal title is "2: Python". The terminal content shows a user prompt, a command to run a Python script, and the script's output. The script calculates the area of a regular polygon with 5 sides of length 5. The output is 43.01193501472417. The terminal window has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, and Settings. The status bar at the bottom shows "master*", "Python 3.8.5 64-bit", "0 0", "Ln 12, Col 1 (317 selected)", "Spaces: 4", "UTF-8", "LF", "Python", and a bell icon.

```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_4.py
enter the number of sides in your regular polygon :    5
enter the length of a side:    5
the area of the regular polygon is 43.01193501472417
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

master* Python 3.8.5 64-bit 0 0 Ln 12, Col 1 (317 selected) Spaces: 4 UTF-8 LF Python

05. write a program to edit the phone number of arvind in file phonebook.txt

```
import re , os

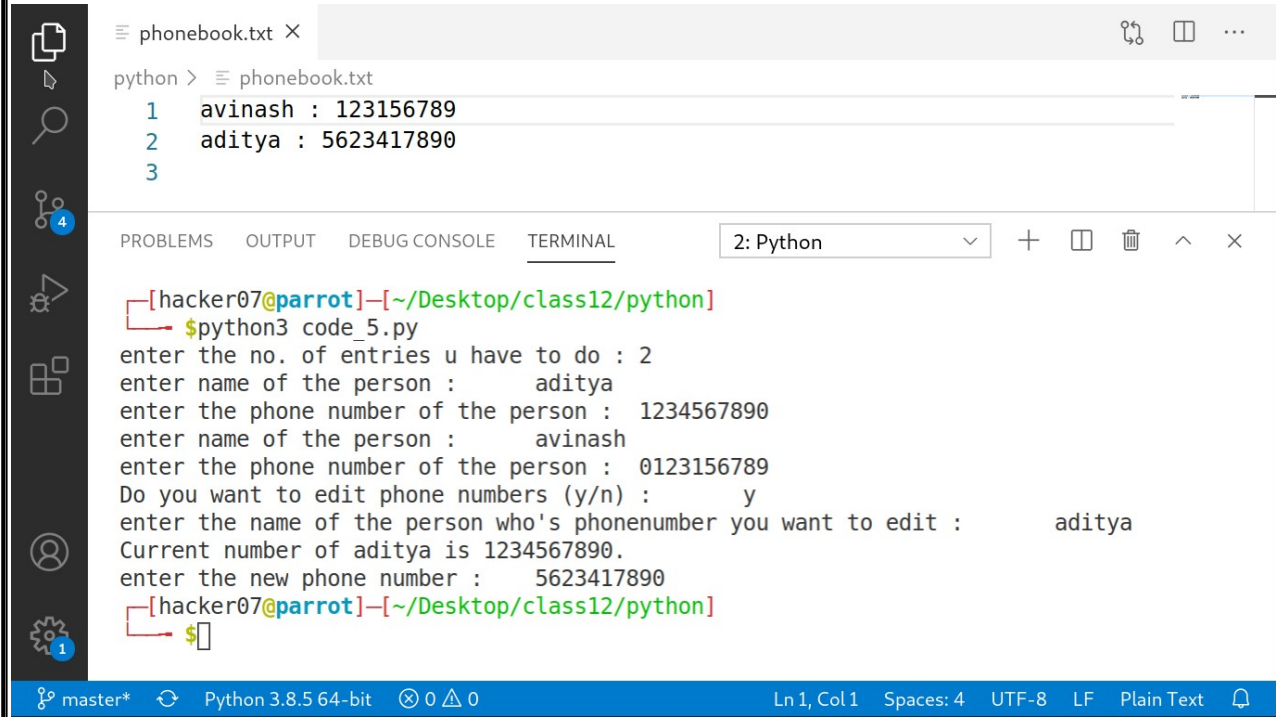
with open("phonebook.txt", "w") as f:

no_of_entries = int(input("enter the no. of entries u have to do :\n\t"))

for _ in range(no_of_entries):
    entry1 = input("enter name of the person :\t")
    entry2 = int(input("enter the phone number of the person :\t"))
    entry = (f"{entry1} : {entry2}\n")
    f.write(entry)
    f.close()
    permmission = input("Do you want to edit phone numbers (y/n) :\t")

    if permmission == "y":
        name = input("enter the name of the person who's phonenumber\nyou want to edit :\t")
        with open("phonebook.txt", "r+") as f:
            for line in f:
                if name in line:
                    a = line
                    current_ph_no = re.compile(r'\d\d\d\d\d\d\d\d\d\d')
                    mo = current_ph_no.search(a)
                    old_number = mo.group()
                    print(f"Current number of {name} is {old_number}.")
                    new_entry = int(input("enter the new phone number :\t"))
                    new_no = f"{name} : {new_entry}\n"
                    f.writelines(new_no)
                    with open("phonebook.txt", "r") as f:
                        data = f.readlines()
                    with open("phonebook.txt", "w") as f:
                        for line in data:
                            if line.strip("\n") != f"{name} : {old_number}":
                                f.write(line)
                        else:
                            print("thank you")
                            exit()
```

Output:-



The screenshot shows a code editor with a file named 'phonebook.txt' open. The editor contains a Python script with three lines of code. Below the code editor is a terminal window showing the execution of the script. The terminal output shows the script running successfully, prompting the user to enter the number of entries, the name and phone number of each person, and whether to edit the phone numbers. The script then displays the current phone number for 'aditya' and prompts the user to enter a new phone number.

```
python > phonebook.txt
1 avinash : 123156789
2 aditya : 5623417890
3

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 2: Python
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_5.py
enter the no. of entries u have to do : 2
enter name of the person : aditya
enter the phone number of the person : 1234567890
enter name of the person : avinash
enter the phone number of the person : 0123156789
Do you want to edit phone numbers (y/n) : y
enter the name of the person who's phonenumber you want to edit : aditya
Current number of aditya is 1234567890.
enter the new phone number : 5623417890
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

master* Python 3.8.5 64-bit 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text

06. write a program to display bar graph between person and score

```
import numpy as np
import matplotlib.pyplot as plt

# data to plot

n_groups = 4
aditya = (90, 55, 40, 65)
avinash = (85, 62, 54, 20)

# create plot

index = np.arange(n_groups)
bar_width = 0.35
opacity = 0.8

rects1 = plt.bar(index, aditya, bar_width,
alpha = opacity,
color = 'b',
label = 'Aditya')

rects2 = plt.bar(index, avinash, bar_width,
alpha = opacity,
color = 'r',
label = 'Avinash')

plt.xlabel('Person')
plt.ylabel('Scores')
plt.title('Scores by person')
plt.xticks(index + bar_width, ('A', 'B', 'C', 'D'))
plt.legend()

plt.tight_layout()
plt.show()
```

Output:-



07. write a program to display line graph between person and score

```
import numpy as np
import matplotlib.pyplot as plt

# data to plot
n_group = 4
aditya = (90, 55, 40, 65)
avinash = (85, 62, 54, 20)

# create plot
index = np.arange(n_group)
bar_width = 0.35
opacity = 0.8

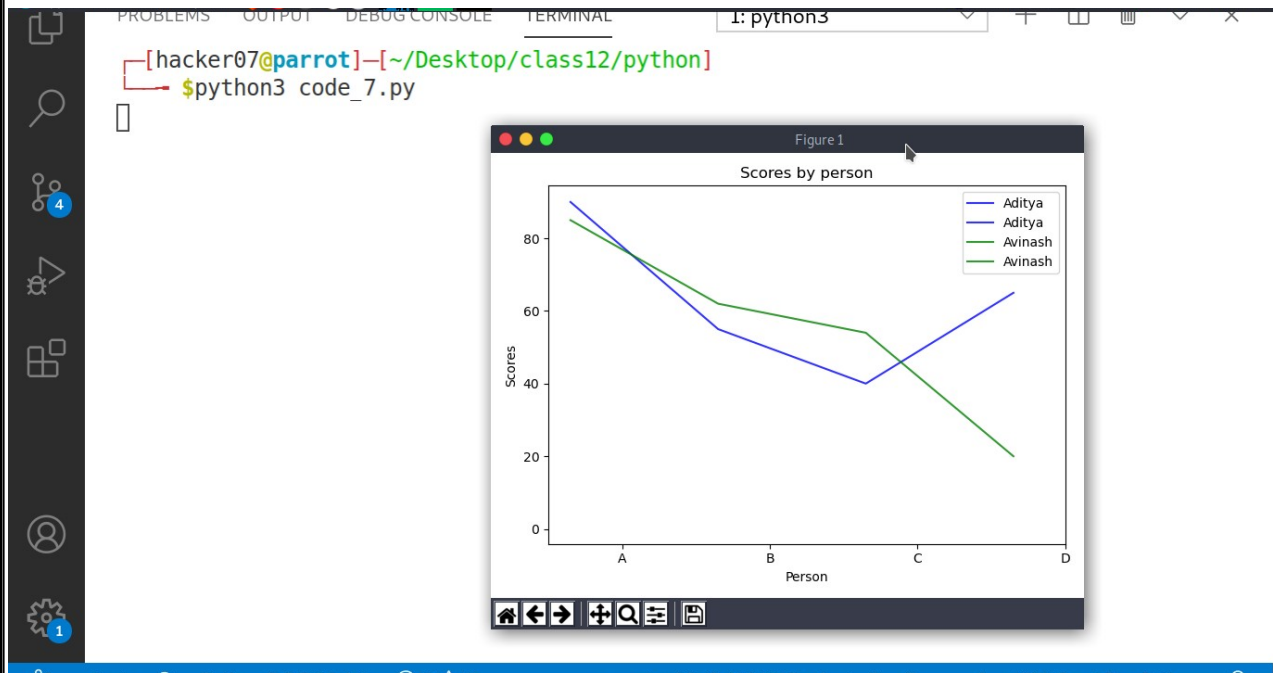
rects1 = plt.plot(index, aditya, bar_width,
alpha = opacity,
color = 'b',
label = 'Aditya')

rects2 = plt.plot(index, avinash, bar_width,
alpha = opacity,
color = 'g',
label = 'Avinash')

plt.xlabel('Person')
plt.ylabel('Scores')
plt.title('Scores by person')
plt.xticks(index + bar_width, ('A', 'B', 'C', 'D'))
plt.legend()

plt.tight_layout()
plt.show()
```

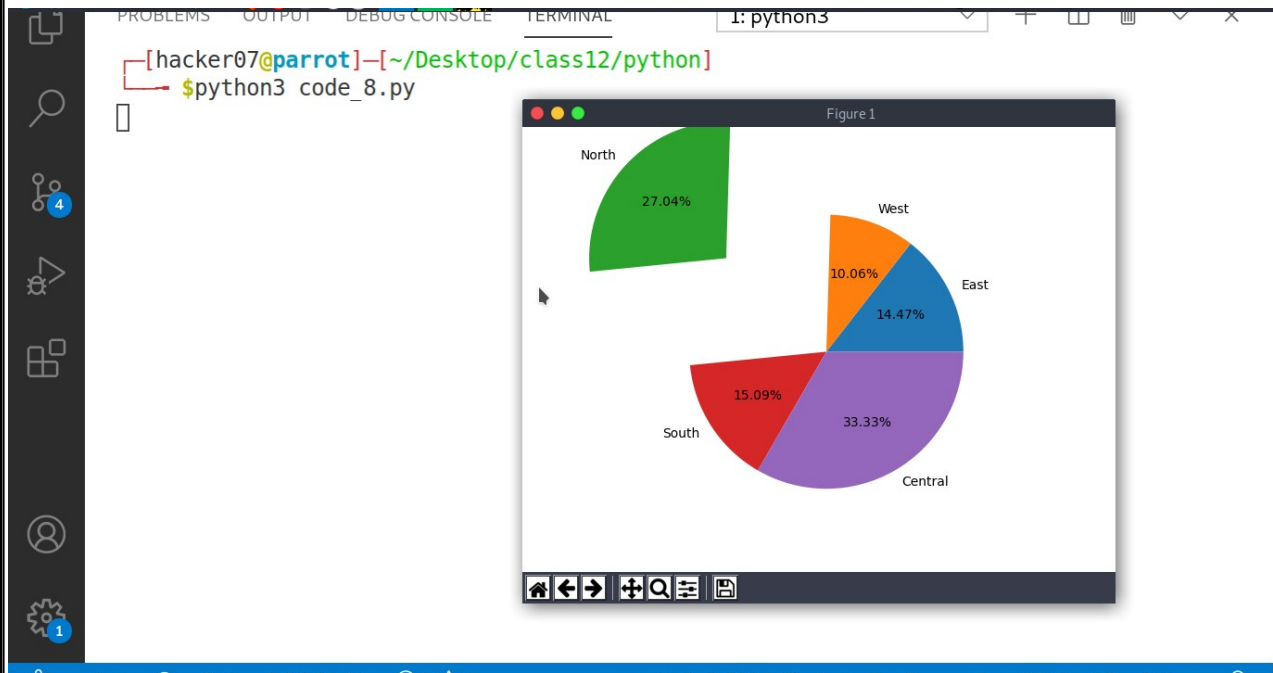
Output:-



08. write a program to display a pie chart

```
import matplotlib.pyplot as plt
c = (23, 16, 43, 24, 53)
zones = ('East', 'West', 'North', 'South', 'Central')
plt.axis("equal")
plt.pie(c, labels=zones, explode=[0,0,1,0,0], autopct="%1.2f%%")
plt.show()
```

Output:-



09. write a program to add a element at end of a list only if element is not in list.

```
a = [1, 7, 5, 25, 78, 38]
```

```
while True:
```

```
try:
```

```
input0 = int(input("Enter a number between 0-100 :\t"))
```

```
except:
```

```
print("Invalid entery")
```

```
if input0 in a:
```

```
print("The element you want to enter already exists in the list. Try  
again!!")
```

```
else:
```

```
a.append(input0)
```

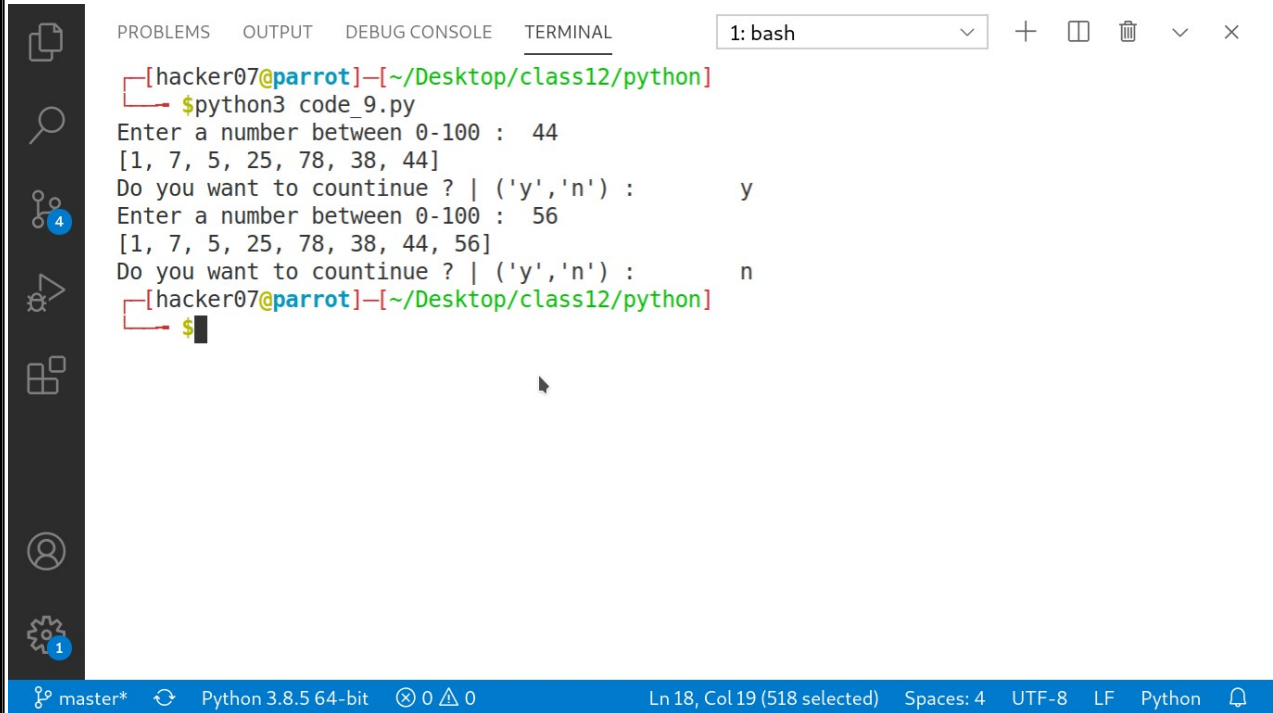
```
print(a)
```

```
decision = input("Do you want to countinue ? | ('y','n') :\t")
```

```
if decision == 'n':
```

```
exit()
```

Output:-



The screenshot shows a VS Code interface with a terminal window open. The terminal title bar indicates the shell is '1: bash'. The terminal content shows the execution of a Python script named 'code_9.py'. The script prompts the user to enter a number between 0-100. The first input is 44, and the second is 56. The script then asks if the user wants to continue, with 'y' and 'n' as options. The terminal output is as follows:

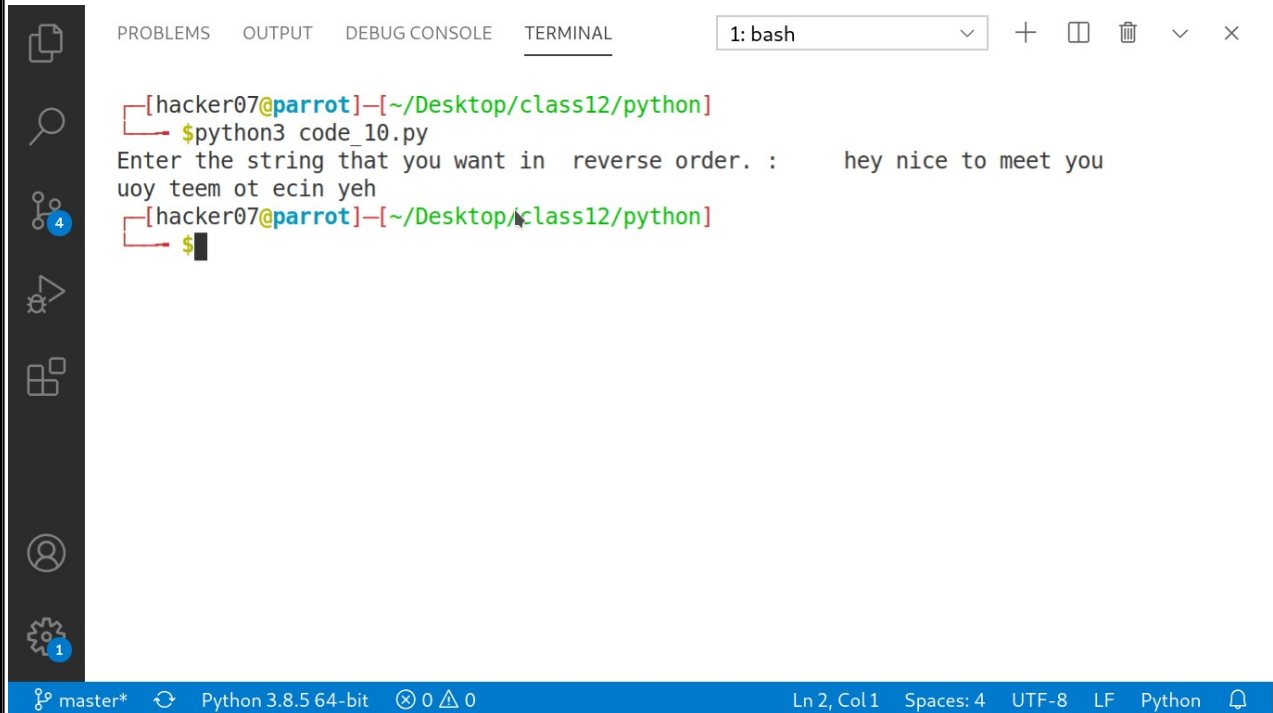
```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_9.py
Enter a number between 0-100 : 44
[1, 7, 5, 25, 78, 38, 44]
Do you want to countinue ? | ('y','n') : y
Enter a number between 0-100 : 56
[1, 7, 5, 25, 78, 38, 44, 56]
Do you want to countinue ? | ('y','n') : n
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

The status bar at the bottom of the VS Code window shows the following information: 'master*' (branch), 'Python 3.8.5 64-bit' (interpreter), '0 0' (errors/warnings), 'Ln 18, Col 19 (518 selected)' (cursor position), 'Spaces: 4' (indentation), 'UTF-8' (encoding), 'LF' (line endings), 'Python' (language), and a bell icon for notifications.

10. write a program to show string in a reverse order.

```
string = input("Enter the string that you want in reverse order. :\t")  
print(string[::-1])
```

Output:-



The image shows a screenshot of a Visual Studio Code (VS Code) terminal window. The terminal is titled "1: bash" and shows the following output:

```
[hacker07@parrot]--[~/Desktop/class12/python]  
$python3 code_10.py  
Enter the string that you want in reverse order. :    hey nice to meet you  
uoy teem ot ecin yeh  
[hacker07@parrot]--[~/Desktop/class12/python]  
$
```

The terminal window has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Accounts, and Settings. The status bar at the bottom shows "master*", "Python 3.8.5 64-bit", "0 0", "Ln 2, Col 1", "Spaces: 4", "UTF-8", "LF", "Python", and a bell icon.

11. write a python code to demonstrate working of binary search in library

```
data = []
input0 = int(input("How much numbers you want to enter :\t"))
for _ in range(input0):
    input1 = int(input("Enter the no to the list :\t"))
    data.append(input1)
print(data)
print("List data is being sorted .....")
data.sort()
print(f"Sorted data is {data}.")

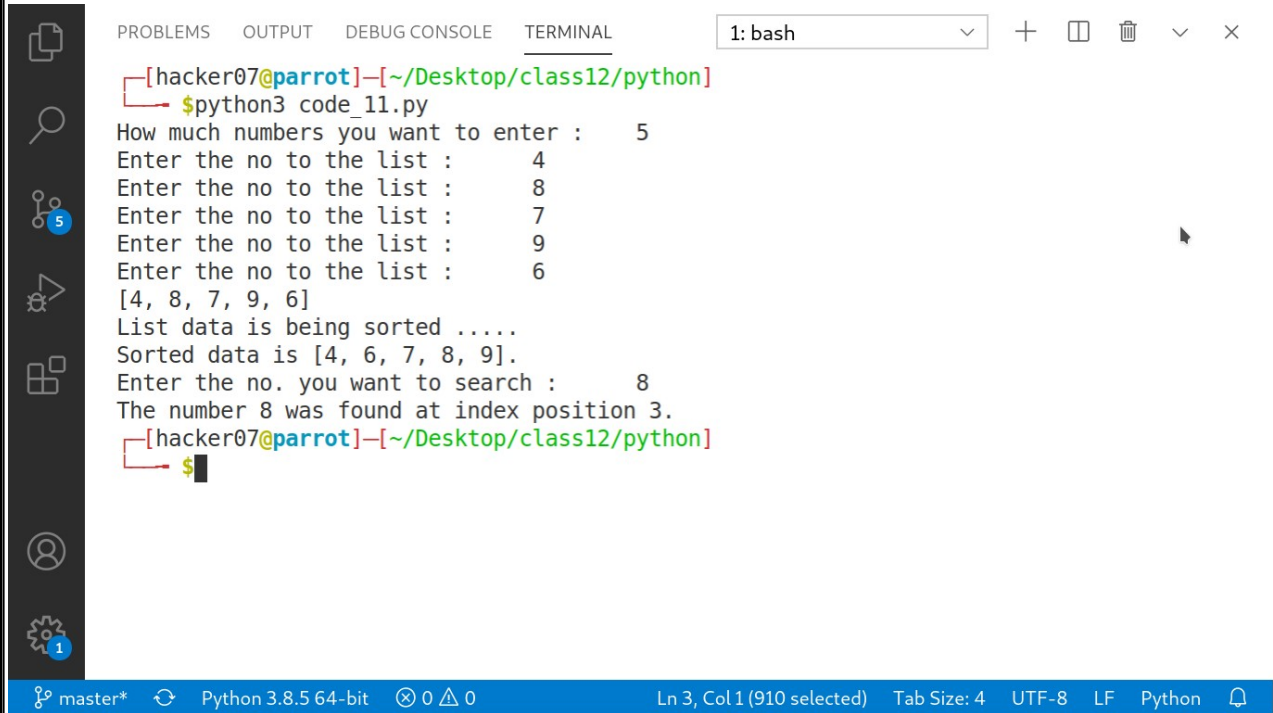
def findValue(data, number_to_find, low, high):
    if high >= low:
        middle = low + (high - low) // 2

        if data[middle] == number_to_find:
            return middle
        elif data[middle] < number_to_find:
            return findValue(data, number_to_find, middle + 1, high)
        else:
            return findValue(data, number_to_find, low, middle - 1)
        else:
            return -1

number_to_find = int(input("Enter the no. you want to search :\t"))
final = findValue(data, number_to_find, 0, len(data) - 1)

if final == -1:
    print("This item was not found in the list.")
else:
    print("The number " + str(number_to_find) + " was found at index position " + str(final) + ".")
```

Output:-



The screenshot shows a VS Code interface with a terminal window open. The terminal title bar indicates it is a bash shell. The prompt is [hacker07@parrot]~/. Desktop/class12/python. The user has executed the command \$python3 code_11.py. The output of the script is as follows:

```
[hacker07@parrot]~/. Desktop/class12/python
$python3 code_11.py
How much numbers you want to enter :    5
Enter the no to the list :             4
Enter the no to the list :             8
Enter the no to the list :             7
Enter the no to the list :             9
Enter the no to the list :             6
[4, 8, 7, 9, 6]
List data is being sorted .....
Sorted data is [4, 6, 7, 8, 9].
Enter the no. you want to search :      8
The number 8 was found at index position 3.
[hacker07@parrot]~/. Desktop/class12/python
$
```

The VS Code status bar at the bottom shows the current file is master*, the interpreter is Python 3.8.5 64-bit, and the cursor is at Ln 3, Col 1 (910 selected). The tab size is 4, and the encoding is UTF-8 with LF line endings.

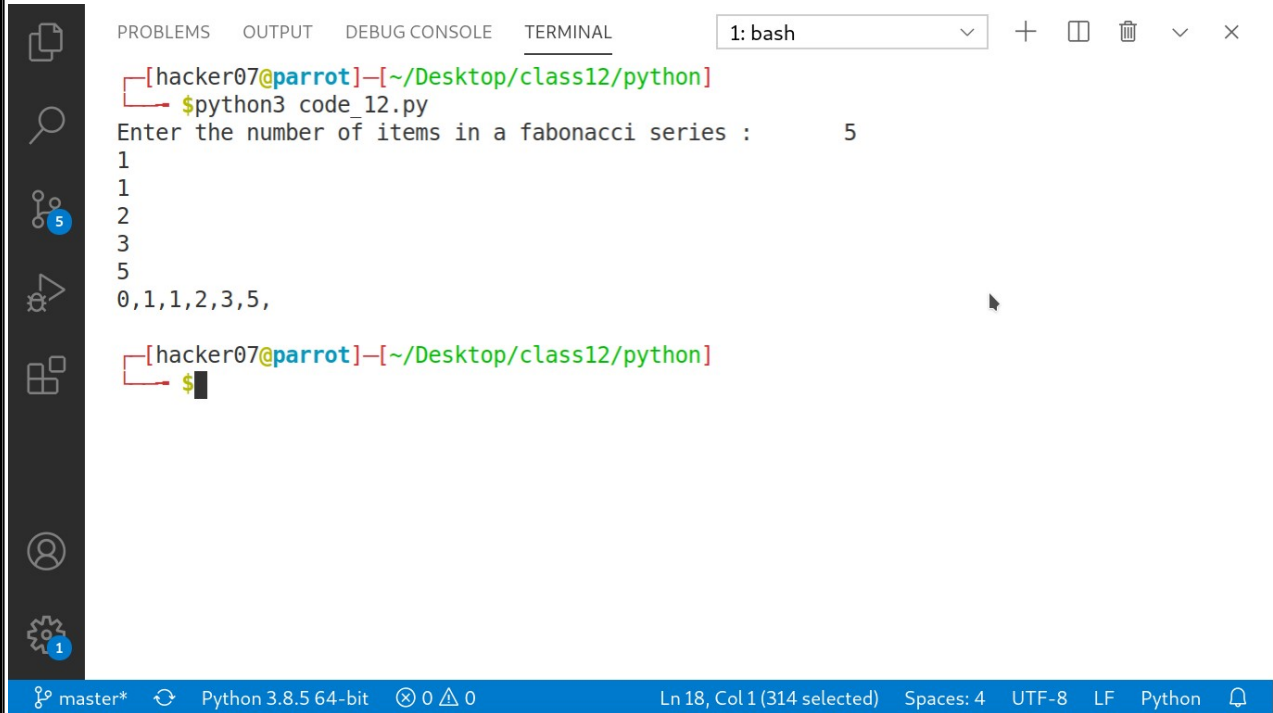
12. write a program for fabonacci series.

```
a = 0
b = 1
fabonacci_series = [0]

c = int(input("Enter the number of items in a fabonacci series :\t"))
for _ in range(c):
    d = a + b
    a = b
    b = d
    print(a)
    fabonacci_series.append(a)

for items in fabonacci_series:
    print(items, end=",")
    print("\n")
```

Output:-



The screenshot shows a VS Code interface with a terminal window open. The terminal title bar indicates the shell is '1: bash'. The terminal content shows a user prompt '[hacker07@parrot]~' followed by the command '\$python3 code_12.py'. The program prompts 'Enter the number of items in a fabonacci series : 5'. It then outputs the numbers 1, 1, 2, 3, 5, followed by a comma and a space. The status bar at the bottom shows 'master*' as the active branch, 'Python 3.8.5 64-bit' as the interpreter, and 'Ln 18, Col 1 (314 selected)' as the current cursor position.

```
[hacker07@parrot]~[~/Desktop/class12/python]
$python3 code_12.py
Enter the number of items in a fabonacci series :      5
1
1
2
3
5
0,1,1,2,3,5,
[hacker07@parrot]~[~/Desktop/class12/python]
$
```

master* Python 3.8.5 64-bit 0 0 Ln 18, Col 1 (314 selected) Spaces: 4 UTF-8 LF Python

13. Write a program to find factorial of a number.

```
number = int(input("Enter the number whose factorial you want :\n"))
```

```
factorial = 1
```

```
# check if the number is negative, positive or zero
```

```
if number < 0:
```

```
    print("Sorry, factorial does not exist for negative numbers")
```

```
elif number == 0:
```

```
    print("The factorial of 0 is 1")
```

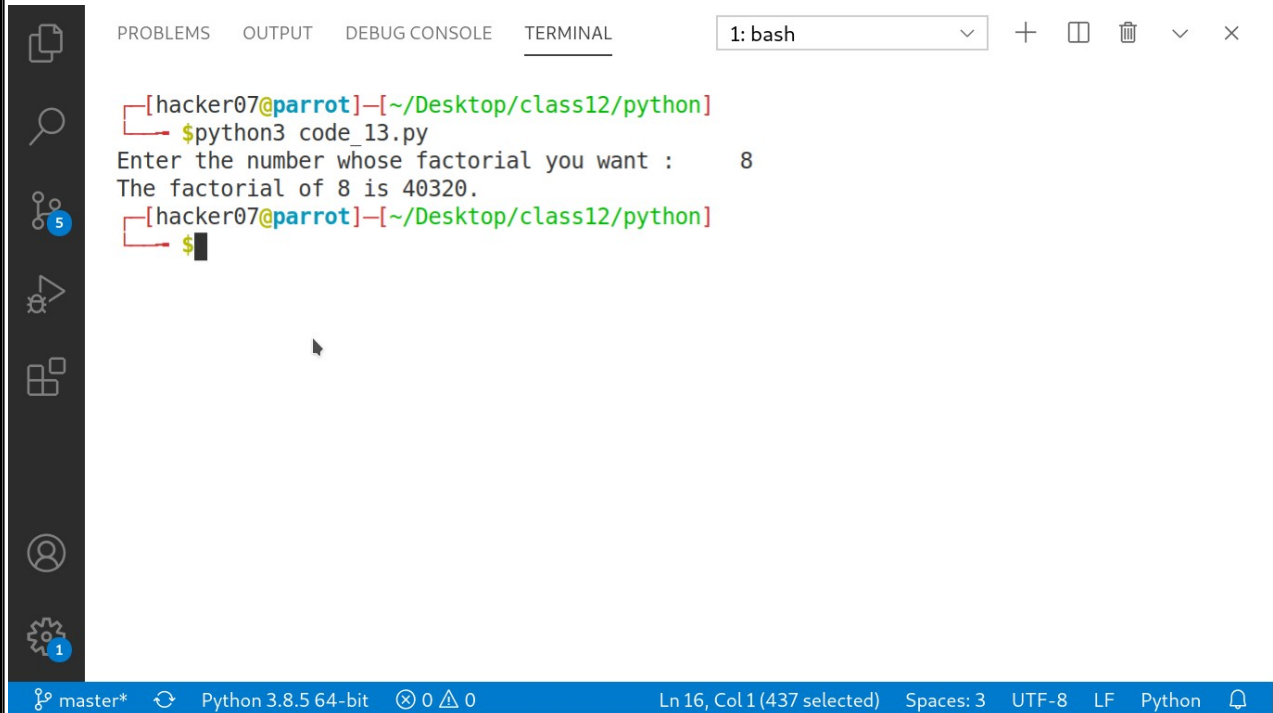
```
else:
```

```
    for i in range(1,number + 1):
```

```
        factorial *= i
```

```
    print(f"The factorial of {number} is {factorial}.")
```

Output:-



The screenshot shows a VS Code interface with the 'TERMINAL' tab active. The terminal output shows a user running a Python script to calculate the factorial of 8, resulting in 40320.

```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_13.py
Enter the number whose factorial you want :      8
The factorial of 8 is 40320.
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

The status bar at the bottom indicates the file is 'master*', the interpreter is 'Python 3.8.5 64-bit', and the cursor is at 'Ln 16, Col 1 (437 selected)'.

14. Write a python program showing practical example of absolute path

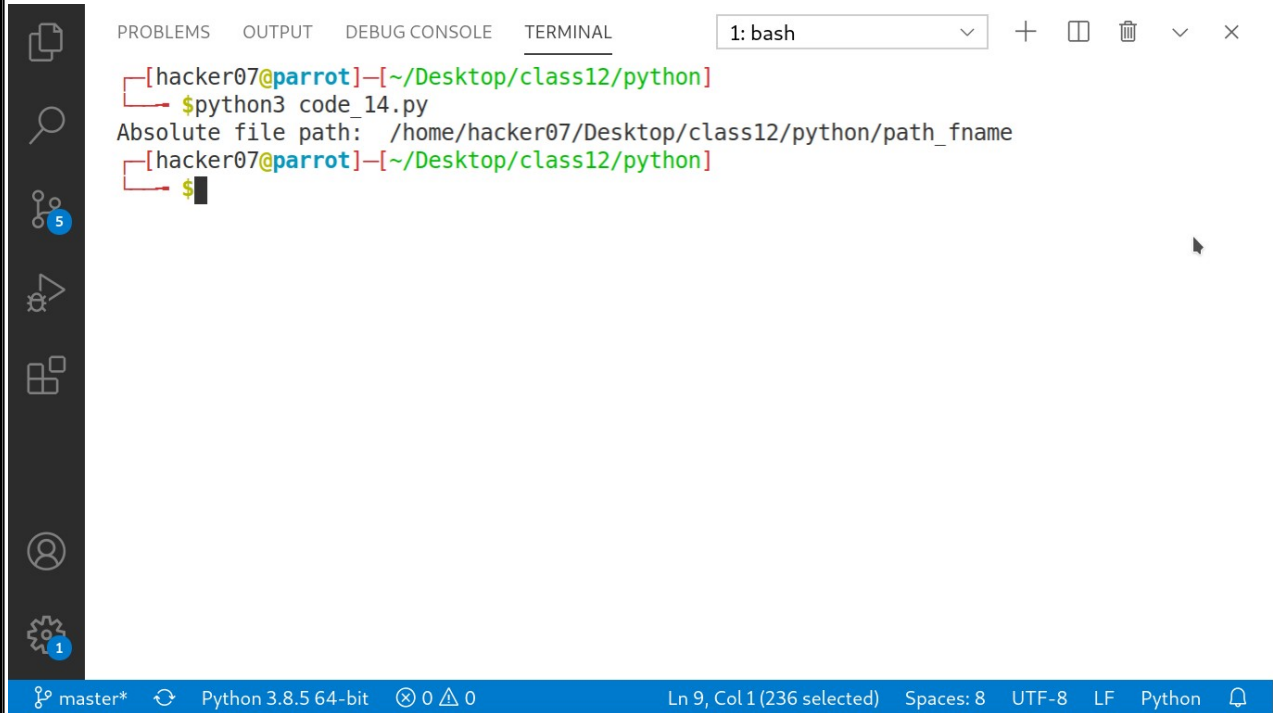
```
import os
```

```
def absolute_file_path(path_fname):
```

```
    return os.path.abspath('path_fname')
```

```
print("Absolute file path: ",absolute_file_path("phonebook.txt"))
```

Output:-



The screenshot shows a VS Code interface with a terminal window open. The terminal title bar indicates it is a 'bash' shell. The terminal content shows a user prompt, a command to run a Python script, the script's output, and a subsequent prompt. The status bar at the bottom provides details about the current file and editor settings.

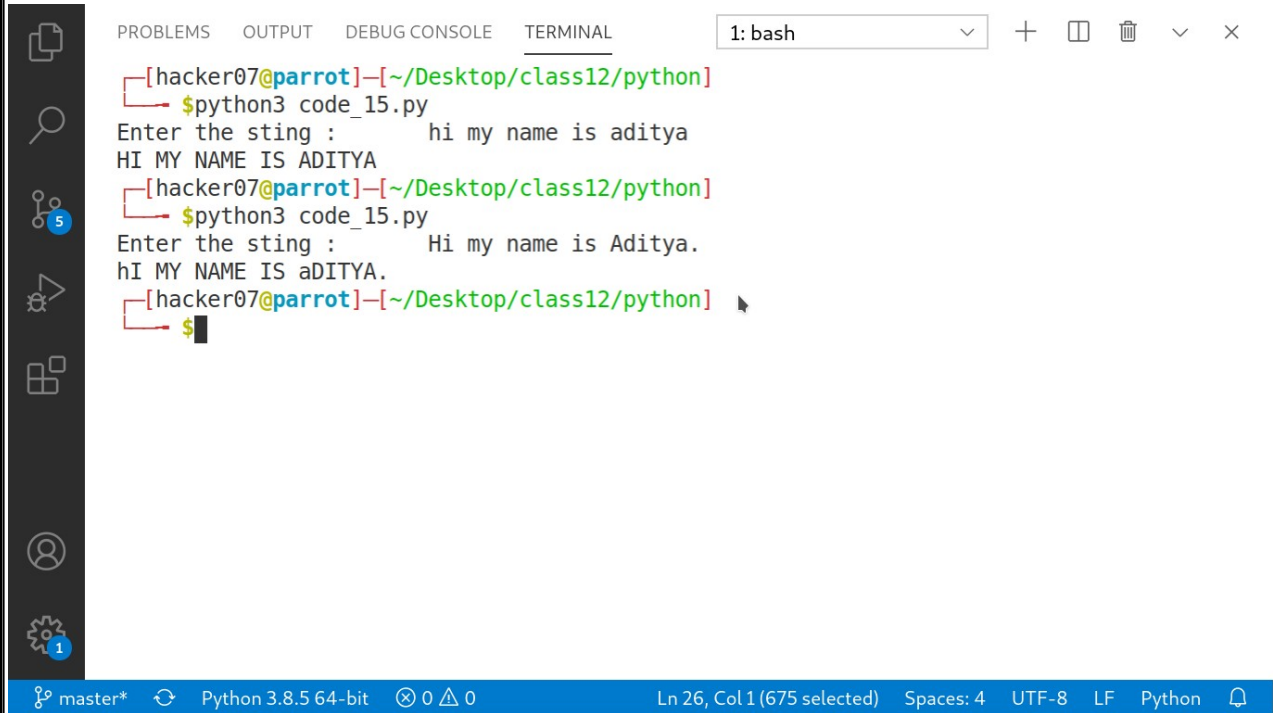
```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_14.py
Absolute file path: /home/hacker07/Desktop/class12/python/path_fname
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

VS Code status bar: master* Python 3.8.5 64-bit 0 0 Ln 9, Col 1 (236 selected) Spaces: 8 UTF-8 LF Python

15. write a program that reads character one by one and store them in UPPER if in upper case or in LOver if in other.

```
def convertOpposite(str):
    ln = len(str)
    for i in range(ln):
        if str[i] >= 'a' and str[i] <= 'z':
            # Convert lowercase to uppercase
            str[i] = chr(ord(str[i]) - 32)
        elif str[i] >= 'A' and str[i] <= 'Z':
            # Convert lowercase to uppercase
            str[i] = chr(ord(str[i]) + 32)
    # Driver code
    if __name__ == "__main__":
        str = input("Enter the sting :\t")
        str = list(str)
        # Calling the Function
        convertOpposite(str)
        str = ''.join(str)
        print(str)
```

Output:-



The screenshot shows a VS Code terminal window with the following content:

```
PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL 1: bash
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_15.py
Enter the sting :      hi my name is aditya
HI MY NAME IS ADITYA
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_15.py
Enter the sting :      Hi my name is Aditya.
hI MY NAME IS aDITYA.
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

The terminal window has a sidebar on the left with icons for Explorer, Search, Source Control, Run and Debug, Extensions, Accounts, and Settings. The status bar at the bottom shows the current file is 'master*', the Python version is 'Python 3.8.5 64-bit', and the cursor is at 'Ln 26, Col 1 (675 selected)'.

16. Write a program for sql connection.

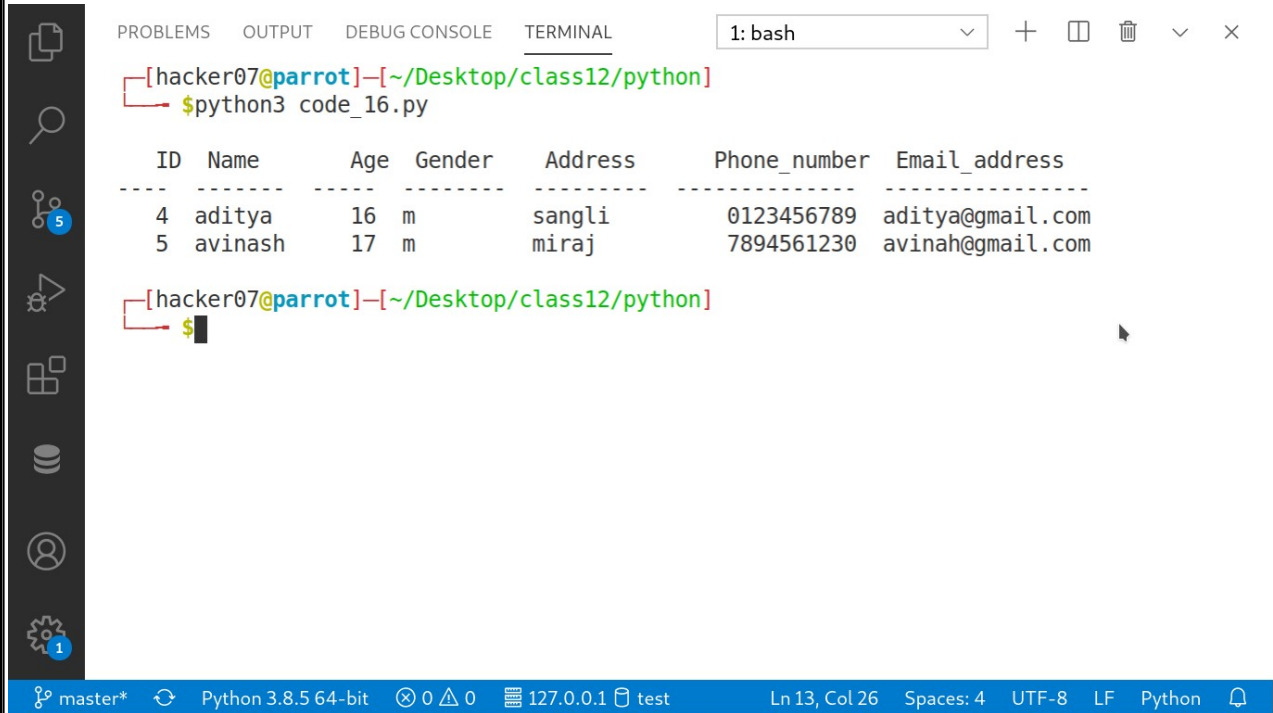
```
import mysql.connector
from tabulate import tabulate

mysqldb = mysql.connector.connect(
    host="localhost",
    user="hacker07",
    passwd="admin1234",
    db = "test"
)

cursor = mysqldb.cursor()
cursor.execute("SELECT * FROM test.student_master")

rows = cursor.fetchall()
print("\n",tabulate(rows, headers=cursor.column_names),"\n")
```

Output:-



```
[hacker07@parrot]--[~/Desktop/class12/python]
$python3 code_16.py
```

ID	Name	Age	Gender	Address	Phone_number	Email_address
4	aditya	16	m	sangli	0123456789	aditya@gmail.com
5	avinash	17	m	miraj	7894561230	avinah@gmail.com

```
[hacker07@parrot]--[~/Desktop/class12/python]
$
```

master* Python 3.8.5 64-bit 0 127.0.0.1 test Ln 13, Col 26 Spaces: 4 UTF-8 LF Python

SQL

-- 17. Create table and insert data into it

```
CREATE TABLE if not exists student(  
ID int Primary Key NOT NULL,  
Name varchar(30) NOT NULL,  
Age int NOT NULL,  
Gender char(1) NOT NULL,  
Address varchar(50) NOT NULL,  
Phone_number char(10) ,  
Email_address varchar(20) ,  
CONSTRAINT chk_stu CHECK (Age<=20 AND Gender in ('m','f','o'))  
);
```

```
INSERT INTO test.student VALUES  
(01,'Jhonny',17,'m','sangli','1452367895','jhonny@gmail.com');  
INSERT INTO test.student VALUES  
(02,'arvind',16,'m','miraj','7485425614','arvind@gmail.com');  
INSERT INTO test.student VALUES  
(03,'saloni',18,'f','ashta','8547956514','saloni@gmail.com');  
INSERT INTO test.student VALUES  
(04,'jackson',17,'m','bedag','1578469853','jackson@gmail.com');
```

```
SELECT * FROM test.student;
```

```
CREATE TABLE if not exists student_marks(  
ID int primary key NOT NULL,  
Phy_marks int NOT NULL,  
chem_marks int NOT NULL,  
maths_marks int NOT NULL,  
);
```

```
INSERT INTO test.student_marks VALUES (01,92);  
INSERT INTO test.student_marks VALUES (02,95);  
INSERT INTO test.student_marks VALUES (03,87);  
INSERT INTO test.student_marks VALUES (04,85);
```

```
SELECT * FROM test.student_marks;
```

Output:-

student@test X

SELECT * FROM test.student;

Input To Search Data

#	ID	Name	Age	Gend	Addr	Phone_nu	Email_address
0	1	Jhonny	17	m	sangli	1452367895	jhonny@gmail.com
1	2	arvind	16	m	miraj	7485425614	arvind@gmail.com
2	3	saloni	18	f	ashta	8547956514	saloni@gmail.com
3	4	jackson	17	m	bedag	1578469853	jackson@gmail.com

master* Python 3.8.5 64-bit 0 0 127.0.0.1 test student Row 3, Col 8 2ms

student_marks@test X

SELECT * FROM test.student_marks;

Input To Search Data

#	ID	Mark
0	1	92
1	2	95
2	3	87
3	4	85

master* Python 3.8.5 64-bit 0 0 127.0.0.1 test student_marks Row 3, Col 3 3ms

-- 18. Write a SQL program for joining two tables using select statement

```
select student.Name, student_marks.marks  
from student, student_marks  
where student.ID=student_marks.ID;
```

Output:-

The screenshot shows a web application interface with a dark sidebar on the left containing icons for file management, search, data, and settings. The main area has a header 'student@test' and a text box containing a SQL query: `select student.Name, student_marks.marks from student, student_marks where student.ID=student_marks.ID;`. Below the query is an 'Input To Search Data' field and a row of action buttons (share, zoom, search, refresh, undo, redo, play). The results are displayed in a table with columns '#', 'Name', and 'mark:'. The table contains four rows of data. At the bottom, a status bar shows 'master*', 'Python 3.8.5 64-bit', '0 0', '127.0.0.1 test', 'student', 'Row 3, Col 3', and '191ms'.

```
select student.Name,
student_marks.marks
from student, student_marks
where student.ID=student_marks.ID;
```

Input To Search Data

#	Name	mark:
0	Jhonny	92
1	arvind	95
2	saloni	87
3	jackson	85

master* Python 3.8.5 64-bit 0 0 127.0.0.1 test student Row 3, Col 3 191ms

-- 19. Joining two tables using select statement and applying arithmetic operations

```
select student.Name, student_marks.Phy_marks,  
student_marks.chem_marks, student_marks.maths_marks,  
student_marks.Phy_marks + student_marks.chem_marks  
+ student_marks.maths_marks as 'Total'  
from student, student_marks  
where student.ID=student_marks.ID;
```

Output:-

The screenshot displays the Visual Studio Code (VS Code) interface. On the left, the Explorer sidebar shows a project structure with a folder named 'CLASS12' containing a 'python' subfolder with files 'code_1.py' through 'code_16.py'. The main editor window, titled 'student@test X', contains a SQL query in a text area:

```
select student.Name,  
student_marks.Phy_marks,  
student_marks.chem_marks,  
student_marks.maths_marks,  
student_marks.Phy_marks +
```

Below the query is a search bar labeled 'Input To Search Data' and a row of action buttons (blue, grey, white, green, orange, red). The output of the query is displayed as a table with the following data:

	#	Name	Phy	chem	maths	Total
	0	Jhonny	92	52	58	202
	1	arvind	95	57	26	178
	2	saloni	87	78	89	254

The status bar at the bottom indicates the current environment: 'master*' (Python 3.8.5 64-bit), '127.0.0.1 test', 'student', 'Row 2, Col 6', and '4ms'.

-- 20. Joining two tables using select statment and apllying Order by clause.

```
select student.Name, student_marks.Phy_marks,  
student_marks.chem_marks,  
student_marks.maths_marks,student_marks.Phy_marks +  
student_marks.chem_marks +student_marks.maths_marks as  
'Total'  
from student, student_marks  
where student.ID=student_marks.ID  
order by Total asc;
```

Output:-

The screenshot displays a Jupyter Notebook environment. At the top, a terminal window shows a SQL query: `student_marks.chem_marks, student_marks.maths_marks, student_marks.Phy_marks + student_marks.chem_marks + student_marks.maths_marks as 'Total' from student, student_marks`. Below the query, a table with 7 columns is displayed: #, Name, Phy, chem, math, and Total. The table contains three rows of data for students arvind, Jhonny, and saloni. The bottom status bar indicates the environment is Python 3.8.5 64-bit, running on a test environment with IP 127.0.0.1, and the current cell is at Row 2, Col 6, taking 45ms to execute.

```
student_marks.chem_marks,
student_marks.maths_marks, student_marks.Phy_ma
rks + student_marks.chem_marks
+ student_marks.maths_marks as 'Total'
from student, student_marks
```

#	Name	Phy	chem	math	Total
0	arvind	95	57	26	178
1	Jhonny	92	52	58	202
2	saloni	87	78	89	254

master* Python 3.8.5 64-bit 0 0 127.0.0.1 test student Row 2, Col 6 45ms

-- 20. Joining two tables using select statement and applying GROUP BY clause.

```
select student.ID, student.Name  
from student  
GROUP BY gender = 'm' or 'f'
```

Output:-

The screenshot shows a web application interface with a dark sidebar on the left containing various icons. The main area has a header bar with the text "student@test X" and a search icon. Below the header, there is a text input field containing the SQL query: `select student.ID, student.Name
from student
GROUP BY gender = 'm' or 'f'`. Below the query field is a search bar labeled "Input To Search Data" and a row of six colored buttons: blue, grey, white, green, orange, and red. Below these buttons is a table with the following data:

	#	ID	Name
<input type="checkbox"/>	0	3	saloni
<input type="checkbox"/>	1	1	Jhonny

At the bottom of the interface is a blue status bar with the following text: "master* Python 3.8.5 64-bit 0 0 127.0.0.1 test student Row 1, Col 3 2ms".

-----!!! Thank you !!!-----