```
Python program 1 to 20
#1
#Write a program to purposefully raise Indentation Error and correct
it
a=int(input("enter first number"))
b=int(input("enter second number"))
sum=a+b
  print(f"addition of {a} and {b} is", sum)
print(f"addition of {a} and {b} is", sum)


#2
"""Write a program to compute distance between two point staking
input
from the user (Pythagorean Theorem)"""
a=int(input("enter side a"))
b=int(input("enter side b"))
#c=(a2+b2)/0.5
c=(a*2+b*2)/0.5
print("shortest distance between two points is", c)


#3
"""Write a program add.py that takes 2 numbers as command line
arguments
and prints its sum."""
import sys
num1 = float(sys.argv[1])
num2 = float(sys.argv[2])
print("Sum:", num1 + num2)

#4
#Python program to swap two variables
a=int(input("enter a no"))
b=int(input("enter a no"))
a=b+a
b=a-b
a=a-b
print("swap two variable",a,b)


#5
#Write a program to verify the number is odd, even or prime
```

```python
a=int(input("enter a no"))
if a%2==0:
    print("number is even")
else:
    print("number is odd")

if a==1:
    print("number is not prime")
elif a>1:
    for i in range(2,a):
        if a%i==0:
            print("number is not prime")
            break
    else:
        print("number is prime")
else:
    print("number is not prime")


#6
#Write a program to find out the positive and negative number in
list

n = int(input("Enter the number of elements in the list: "))
list1 = []
for i in range(n):
    element = int(input("Enter element {}: ".format(i + 1)))
    list1.append(element)

positive_nos = []
negative_nos = []

# Separating positive and negative numbers
for num in list1:
    if num >= 0:
        positive_nos.append(num)
    else:
        negative_nos.append(num)

print("List includes all elements:", list1)
print("Positive numbers list:", positive_nos)
```

```
print("Negative numbers list:", negative_nos)
#7
#Write a program to find the median of the tuples.

a = int(input("Enter the range of the list: "))
ls = []
for i in range(a):
  b = int(input("Enter the number: "))
  ls.append(b)
print("Your list is:", ls)

ls.sort()
n = len(ls)
if n % 2 == 0:
 # For even-length lists, calculate the average of the two middle
elements
    c = (ls[n // 2 - 1] + ls[n // 2]) / 2
    print("The median =", c)
else:
 # For odd-length lists, simply take the middle element
    c = ls[n // 2]
    print("Median =", c)


#8
"""Write a program for finding whether the string is palindrome or
not."""
# Take input from the user
string_to_check = input("Enter a string: ")

# Remove spaces and convert to lowercase for case-insensitive
comparison
string_to_check = string_to_check.replace(" ", "").lower()

# Initialize start and end pointers
start = 0
end = len(string_to_check) - 1

# Assume the string is a palindrome initially
is_palindrome = True

# Check characters from start and end, moving towards the middle
```

```
while start < end:
    # If characters at start and end don't match, it's not a
palindrome
    if string_to_check[start] != string_to_check[end]:
        is_palindrome = False
        break

    # Move the start pointer towards the middle
    start += 1

    # Move the end pointer towards the middle
    end -= 1

# Print result based on the value of is_palindrome
if is_palindrome:
    print("Yes, the string is a palindrome.")
else:
    print("No, the string is not a palindrome.")

#9
"""Write a function for breaking the set into the list of the
sets."""
def lis(lst):
    r = []
    for i in lst:
        s = set()
        s.add(i)
        r.append(s)
    return(r)
lst = []
a = int(input("Enter the range of set:"))
for i in range (a):
    b = int(input("Enter the number: "))
    lst.append(b)
print(lis(lst))


#10
"""Write a program for printing the cube of list elements using
lambda"""

a = int(input("Enter the range of list:"))
```

```
num = []
for i in range (a):
 b = int(input("Enter the number:"))
 num.append(b)
cubes = list(map(lambda x: x ** 3, num))
print("Your list elements:",num)
print("Cube of list elements:", cubes)

#11
"""Write a program to find the even number from a list by using the
filter()
function. """

n=int(input("enter number"))
l1=[]
for i in range(n):
    element=int(input("enter the number"))
    l1.append(element)
print("list",l1)

def even(n):
    return n % 2 == 0
eve = list(filter(even, l1))
print("Even numbers from the list:", eve)



#12
"""Write a program in Python that takes the name, roll number, and
marks of a student
and calculates their percentage, and prints the grade to demonstrate
class, object and
methods in Python """

class student:
    def st_info(self,name,rollno):
        self.name=name
        self.rollno=rollno
    def display(self):
        print("name of the student:", self.name)
        print("rollno of the student: ",self.rollno)
```

```
m1=int(input("enter the number"))
m2=int(input("enter the number"))
m3=int(input("enter the number"))
m4=int(input("enter the number"))
t=m1+m2+m3+m4
per=t/4
a=student()
a.st_info("x",1)
a.display()
print("total marks", t)
print("percentage",per)
if per>90 and per<=100:
    print("grade A")
elif per>80 and per<=90:
    print("grade B")
elif per>=70 and per<=80:
    print("grade c")
elif per > 60 and per <=70:
    print("grade D")
elif per>50 and per<=60:
    print("grade E")
elif per>40 and per<=50:
    print("grade F")
else:
    print("fail")

#13
"""Write a function to demonstrate the read standard input
and write to the standard output and standard error streams"""

import sys
sys.stdin=open('data.txt','r')
f=sys.stdin.read()
print(f)
sys.stdin.close()

sys.stdout=open('write.txt','w')
sys.stdout.write("hello world")
sys.stdout.close()

sys.stderr=open('error.txt','w')
sys.stderr.write("this is warning")
```

```
sys.stderr.close()
#14.1
"""Write a program to calculate the average of the n numbers of
arguments
using Command-line Arguments. """

import sys
num = [int(num) for num in sys.argv[1:]]
av= sum(num)/len(num)
print("AVERAGE",av)

#14.2
#Write a program to print your name by using the shell variables.
import os
username = os.environ.get('USERNAME')
if username:
    print("Hello,", username)
else:
    print("Unable to retrieve username. Please ensure the 'USER'
environment variable is set.")
# Check if the 'USER' environment variable is available
username = os.environ.get('USER')
if username:
    print("next,", username)
else:
    print("Unable to retrieve username. Please ensure the 'USER'
environment variable is set.")


#15
(nhi ata)



#16(server)
"""Write a program to communicate with the server
and client system using Socket programming with the Client and General
Socket Methods. """

import socket
a=socket.socket(family=socket.AF_INET,type=socket.SOCK_DGRAM)
a.bind(("localhost",1126))
while True:
    msg,addr=a.recvfrom(1024)
    m=msg.decode()
    print("message from client",m[-1]+":"+m[0:len(m)-1])
```

```
#Send a Response to the Client
    m1=input("enter msg for client"+m[-1]+":"+"")
    m2=str.encode(m1)
    a.sendto(m2,addr)


#16(client)
"""Write a program to communicate with the server
and client system using Socket programming with the Client and General
Socket Methods. """
import socket
a=socket.socket(family=socket.AF_INET,type=socket.SOCK_DGRAM)
while True:
    m=input("enter the message for server:")
    m1=str.encode(m+"1")
    m2=("localhost",1126)
    a.sendto(m1,m2)
#Receive and Print Server's Response
    msg,addr=a.recvfrom(1024)
    m3="message from server:"+msg.decode()
    print(m3)
```

17.

```
#17
"""Write a program to print the odd and even
numbers by using thread.
"""
import threading
def odd():
    for i in range(1,10,2):
        print("odd numbers",i)

def even():
    for i in range(2,10,2):
        print("even numbers",i)
t1=threading.Thread(target=odd)
t2=threading.Thread(target=even)

t1.start()
t2.start()

t1.join()
t2.join()

print("threading done")
```

18.

```
#18
""" Create a function to find the fruits that start
with a specific letter by using a compound data structure. """

def func(fruits,letter):
    letter=letter.lower()
    if letter in fruits:
        return fruits[letter]
    else:
        return[]

f={
    'a':['apple'],
    'b':['blueberry']
}

l="b"
result=func(f,l)
print(f"fruits start with {l}:",result)


#19
#Create a function by using the list, tuples, and dictionaries.

def func(lst,tpl,dct):
    print("list")
    for items in lst:
        print(items)

    print("tuple")
    for items in tpl:
        print(items)

    print("dictionary")
    for key, value in dct.items():
        print(f"{key}: {value}")

list1=[1,'b',3,'a']
tuple1=(1,2,3,4)
d1={"a":1,"b":2}
func(list1,tuple1,d1)
```

```python
#20(a)
#Read and write data from/to files in Python
#20th
file=open('data.txt', 'r')
f=file.read()
print(f)

data = "Hello, World!\nThis is a test."
file=open('output.txt', 'w')
file.write(data)
```

```python
#20(b)
def add_data_to_file(filename, data):
    """Append data to the end of the file."""
    with open(filename, 'a') as file:
        file.write(data + '\n')

def read_specific_data_from_file(filename, line_number):
    """Read specific line from the file."""
    with open(filename, 'r') as file:
        lines = file.readlines()
        return lines[line_number - 1].strip()

# Add data to the file
add_data_to_file("data.txt", "Hello, World!")  # Add the first line
add_data_to_file("data.txt", "This is a test.")  # Add the second
line

# Read specific data from the file
line_number = 2
data = read_specific_data_from_file("data.txt", line_number)
print(f"Data at line {line_number}: {data}")
```