



---

**#1****LIMITED  
EDITION****VOL 274**

# **PRACTICAL PYTHON PROGRAMMING**

**100+ EXAMPLE PROJECTS**

## **"STAY UPDATED, STAY AHEAD"**

TechVidya is ISO Certified 9001:2015 accredited EdTech Company registered with ROC under the companies act 1956, offers self-paced, online and offline programs. TechVidya pedagogy is rooted in the principle that every young mind should be equipped with exceptional knowledge and skills that benefit them in real life.

**YEAR  
2023**

---

Company Info

**TechVidya**  
EdTech Company

Company Website

**techvidya.education**  
Official Website

Company Contact

**+91 83759 66700**  
Official Number



Practice 1: A Program to Detect Positive and Negative Numbers.....	7
Practice 2: A Program to Detect Even and Odd Number.....	8
Practice 3: A Program to Detect the Greatest of Three Numbers.....	9
Practice 4: A Program to Check for Divisibility of a Number.....	10
Practice 5: A Program to Convert from Celsius to Fahrenheit.....	11
Practice 6: A Program to Convert from Fahrenheit to Celsius.....	12
Practice 7: A Program for Simple Thermometer (Celsius/Fahrenheit) .....	13
Practice 8: A Program to Calculate Mass, Density and Volume.....	14
Practice 9: A Program to Detect the Coordinate of a Point.....	16
Practice 10: A Program to Check for Existence of a Triangle.....	18
Practice 11: A Program to Check for Leap Year.....	19
Practice 12: A Program to Check for Existence of a Circle.....	20
Practice 13: A Program to create quadratic Equation.....	21
Practice 14: A Program to Make Guess of Random Number.....	22
Practice 15: A Program to Print Out the ASCII Table.....	23
Practice 16: A Program to Create a Multiplication Table Using While Loop.....	24
Practice 17: A Program to Create Multiplication Table Using for Loop.....	25
Practice 18: A Program to Convert from Base 2 to 9.....	26
Practice 19: A Program to Build a Simple Calculator.....	27
Practice 20: A Program to Detect Number of Digits in an Integer.....	28
Practice 21: A Program to Get Sum and Products of Digits.....	29
Practice 22: A Program to Make a Binary Search of Number in an Array.....	30
Practice 23: A Program to Get Sum of N Series of an Element.....	31
Practice 24: A Program to Get Number of Even and Odd Digits.....	32
Practice 25: A Program to Get Factorial Using a While Loop.....	33
Practice 26: A Program to Get Factorial Using for Loop.....	34
Practice 27: A Program to Create a Fibonacci Sequence.....	35
Practice 28: A Program to Get the Value of Fibonacci Element.....	36

Practice 29: A Program to Get Find the Greatest Common Divisor.....	37
Practice 30: A Program to Get Maximum Value of a Floating-Point Number.....	38
Practice 31: A Program to Detect Prime Numbers.....	39
Practice 32: A Program for Quadratic Equations with Solutions at Specified Range of Coefficient.....	40
Practice 33: A Program to Reverse Numbers.....	42
Practice 34: A Program to Expand Strings of Alphabet.....	43
Practice 35: A Program to Replace a Substring of a String.....	44
Practice 36: A Program to Select Integers from String.....	45
Practice 37: A Program to Sort Words According to Their Length.....	46
Practice 38: A Program to Find the Longest Word in a String.....	47
Practice 39: A Program to Get Percentage of Uppercase and Lowercase.....	48
Practice 40: A Program to Check for String Palindrome.....	49
Practice 41: A Program to Generate Random Numbers Using Arrays.....	50
Practice 42: A Program to Get the Maximum Element in an Array.....	51
Practice 43: A Program to Get the Minimum Element in an Array.....	52
Practice 44: A Program to Get the Number of Even and Odd Numbers.....	53
Practice 45: A Program to Get Positive Numbers Out of Negative Numbers.....	54
Practice 46: A Program to Get Numbers Greater than the Average of an Array.....	55
Practice 47: A Program to Replace List Items With -1, 0, 1.....	56
Practice 48: A Program to Check for File Extension.....	57
Practice 49: A Program to Remove Exclamation Mark (!) from Text.....	58
Practice 50: A Program to get Intersection of List Using for Loop.....	59
Practice 51: A Program for Simple Intersection of List.....	60
Practice 52: A Program for Longest ordered Sequence in Ascending Order.....	61
Practice 53: A Program to Get the Most Occurrent Element.....	62
Practice 54: A Program to Bubble Sort Elements of an Array.....	63
Practice 55: A Program to Sort Array Using Selection Sorting.....	64
Practice 56: A Program to Generate Matrix of Random Numbers.....	65



Practice 57: A Program to Get the Rows and Columns with Maximum Sum of Elements.....	66
Practice 58: A Program to Sum Items in Rows and Columns of Elements.....	68
Practice 59: A Program to Sum Diagonals of a Matrix.....	69
Practice 60: A Program to Interchange the Principal Diagonals of Matrix.....	70
Practice 61: A Program to Sort Columns of Element by Sorting the First Row.....	71
Practice 62: A Program to Check Rows and Columns that Has Particular Element.....	72
Practice 63: A Program to Generate Beautiful Unicode.....	74
Practice 64: A Program to Get Prices of Products.....	75
Practice 65: A Program to Make List of Dictionaries Using 2 Lists.....	76
Practice 66: A Program to Delete Dictionary Item.....	77
Practice 67: A Program to Return Value of 2 Arguments Using Function.....	78
Practice 68: A Program to Fill List.....	79
Practice 69: A Program to Get the Arithmetic Mean of a List.....	80
Practice 70: A Program to Generate Fibonacci Sequence Using Function.....	81
Practice 71: A Program to Get Fibonacci Value Using Recursion.....	82
Practice 72: A Program to Get Factorial Using Recursion.....	83
Practice 73: A Program to Get the LCM.....	84
Practice 74: A Program to Reverse Word Sequence.....	85
Practice 75: A Program to Search for Binary Numbers.....	86
Practice 76: A Program to Make a Ring Shift or Recycle Items of a List.....	87
Practice 77: A Program to Read Text.....	88
Practice 78: A Program to Use Read Method.....	89
Practice 79: A Program to Use ReadLine Method.....	90
Practice 80: A Program to Use ReadLines Method.....	91
Practice 81: A Program to Write to File.....	92
Practice 82: A Program to Read Text from File to Dictionary.....	93
Practice 83: A Program to Count Number of Lines, Words and Letters in a Text File.....	94
Practice 84: A Program to Capture String Errors.....	95

Practice 85: A Program to Check for Non-Existence of Number.....	96
Practice 86: A Program to Display Error Message for Non-Existed Files.....	97
Practice 87: A Program to Get Division by Zero Error.....	98
Practice 88: A Program to Get Index Out of Exception.....	99
Practice 89: A Program to Raise Exceptions.....	100
Practice 90: A Program to Use Classes and Constructor.....	101
Practice 91: A Program to Fill a List with Natural Numbers.....	102
Practice 92: A Program to Fill a List with Random Numbers.....	103
Practice 93: A Program to Group Dictionaries into Lists.....	104
Practice 94: A Program to Unpack Matrix into One Level List.....	105
Practice 95: A Program to Read Data from File into Lines and Characters.....	106
Practice 96: A Program to Calculate the Orbital Speed.....	107
Practice 97: A Program to Get Sum of Random of 3 Digit Numbers.....	108
Practice 98: A Program to Select Even Numbers from List.....	109
Practice 99: A Program to Calculate the Area and Perimeter of Right-Angle Triangle.....	110
Practice 100: A Program to Get Total Surface Area of a Cylinder.....	111
Practice 101: A Complete Program to Build A Working Digital Electronic Calculator.....	112

### Practice 1: A Program to Detect Positive and Negative Numbers

```
x = float(input("Insert any number: "))

if x>0:
    print("This is a POSITIVE number")
elif x < 0:
    print("This is a NEGATIVE number")
else:
    print("The number is ZERO")
```

#### Output 1

```
Insert any number: 56
This is a POSITIVE number
```

#### Output 2

```
Insert any number: -12
This is a NEGATIVE number
```

#### Output 3

```
Insert any number: 0
The number is ZERO
```

## Practice 2: A Program to Detect Even and Odd Number

```
x = int(input("Insert any number: "))

if x%2 == 0:
    print("This is an EVEN number!")
else:
    print("This is an ODD number!")
```

### Output 1

```
Insert any number: 6
This is an EVEN number!
```

### Output 2

```
Insert any number: 5
This is an ODD number!
```

### Output 3

```
Insert any number: 45
This is an ODD number!
```

### Output 4

```
Insert any number: 100
This is an EVEN number!
```



### Practice 3: A Program to Detect the Greatest of Three Numbers

```
x = int(input("Insert first number: "))
y = int(input("Insert second number: "))
z = int(input("Insert third number: "))

print("The maximum number is : ", end="")
if y <= x >= z:
    print(x)
elif x <= y >= z:
    print(y)
elif x <= z >= y:
    print(z)
```

#### Output 1

```
Insert first number: 12
Insert second number: 45
Insert third number: 8
The maximum number is : 45
```

#### Output 2

```
Insert first number: 23
Insert second number: 10
Insert third number: 5
The maximum number is : 23
```

### Practice 4: A Program to Check for Divisibility of a Number

```
x = int(input("Insert the numerator: "))
y = int(input("Insert the denominator: "))

if x%y == 0:
    print(x, " is divisible by ", y)
else:
    print("No! ", x, " is not divisible by ", y)
```

#### Output 1

```
Insert the numerator: 5
Insert the denominator: 7
No!  5  is not divisible by  7
```

#### Output 2

```
Insert the numerator: 6
Insert the denominator: 2
6  is divisible by  2
```

#### Output 3

```
Insert the numerator: 15
Insert the denominator: 6
No!  15  is not divisible by  6
```

### Practice 5: A Program to Convert from Celsius to Fahrenheit

```
x = int(input("Insert a value in Celsius: "))
```

```
x = round(x*(9/5) + 32)
```

```
print(str(x) + 'F')
```

#### Output 1

```
Insert a value in Celsius: 17
```

```
63F
```

#### Output 2

```
Insert a value in Celsius: 50
```

```
122F
```

#### Output 3

```
Insert a value in Celsius: 30
```

```
86F
```

#### Output 4

```
Insert a value in Celsius: 100
```

```
212F
```

### Practice 6: A Program to Convert from Fahrenheit to Celsius

```
x = int(input("Insert a Fahrenheit value: "))
```

```
x = round((x-32)*(5/9))
```

```
print(str(x)+'C')
```

#### Output 1

```
Insert a Fahrenheit value: 32
```

```
0C
```

#### Output 2

```
Insert a Fahrenheit value: 100
```

```
38C
```

#### Output 3

```
Insert a Fahrenheit value: 50
```

```
10C
```

#### Output 4

```
Insert a Fahrenheit value: 212
```

```
100C
```



**Practice 7: A Program for Simple Thermometer (Celsius/Fahrenheit)**

```
x = input("Insert any value of 'C' or 'F' : ")
```

```
unit = x[-1]
```

```
x = int(x[0:-1])
```

```
if unit == 'C' or unit == 'c':
```

```
    x = round(x*(9/5)+32)
```

```
    print(str(x) + 'F')
```

```
elif unit == 'F' or unit == 'f':
```

```
    x = round((x-32)*(5/9))
```

```
    print(str(x) + 'C')
```

**Output 1**

```
Insert any value of 'C' or 'F' : 32f
```

```
0C
```

**Output 2**

```
Insert any value of 'C' or 'F' : 100C
```

```
212F
```

**Output 4**

```
Insert any value of 'C' or 'F' : 0c
```

```
32F
```

### Practice 8: A Program to Calculate Mass, Density and Volume

```
result = 0
mdv = input("Choose one to calculate(m,d,v) : ")

if mdv == 'm':
    d = float(input("Density: "))
    v = float(input("Volume: "))
    result = ("Mass is : " + str(d*v))
elif mdv == 'd':
    m = float(input("Mass: "))
    v = float(input("Volume: "))
    result = ("Density is : " + str(m/v))
elif mdv == 'v':
    m = float(input("Mass: "))
    d = float(input("Density: "))
    result = ("Volume is : " + str(m/d))
print(result)
```

#### Output 1

```
Choose one to calculate(m,d,v) : m
Density: 10
Volume: 5

Mass is : 50.0
```

**Output 2**

Choose one to calculate(m,d,v) : d

Mass: 10

Volume: 5

Density is : 2.0

**Output 3**

Choose one to calculate(m,d,v) : v

Mass: 10

Density: 5

Volume is : 2.0

**Output 4**

Choose one to calculate(m,d,v) : m

Density: 25

Volume: 7

Mass is : 175.0

**Output 5**

Choose one to calculate(m,d,v) : d

Mass: 25

Volume: 7

Density is : 3.5714285714285716

**Output 6**

Choose one to calculate(m,d,v) : v

Mass: 25

Density: 4

Volume is : 6.25

### Practice 9: A Program to Detect the Coordinate of a Point

```
x = float(input("Insert coordinate of point x: "))
y = float(input("Insert coordinate of point y: "))
if x > 0 and y > 0:
    print("The first quadrant")
elif x < 0 and y > 0:
    print("The second quadrant")
elif x < 0 and y < 0:
    print("The third quadrant")
elif x == 0 and y == 0:
    print("Point of origin")
elif x == 0:
    print("x point")
elif y == 0:
    print("y point")
```

#### Output 1

```
Insert coordinate of point x: 1
Insert coordinate of point y: 1
The first quadrant
```

#### Output 2

```
Insert coordinate of point x: -1
Insert coordinate of point y: 1
The second quadrant
```



**Output 3**

Insert coordinate of point x: -1

Insert coordinate of point y: -1

The third quadrant

**Output 4**

Insert coordinate of point x: 0

Insert coordinate of point y: 0

Point of origin



### Practice 10: A Program to Check for Existence of a Triangle

```
print("Insert length of proposed triangle: ")
x = float(input("x = "))
y = float(input("y = "))
z = float(input("z = "))

if x+y>z and x+z>y and y+z>x:
    print("The triangle of xyz exist")
else:
    print("The triangle does not exist")
```

#### Output 1

```
Insert length of proposed triangle:
x = 5
y = 6
z = 7

The triangle of xyz exist
```

#### Output 2

```
Insert length of proposed triangle:
x = 0
y = 3
z = 8

The triangle does not exist
```

### Practice 11: A Program to Check for Leap Year

```
yr = int(input("Insert any year to check for leap year: "))
```

```
if yr%4 == 0:
```

```
    print("This is a leap year!")
```

```
else:
```

```
    print("This is not a leap year!")
```

#### Output 1

```
Insert any year to check for leap year: 2021
```

```
This is not a leap year!
```

#### Output 2

```
Insert any year to check for leap year: 2022
```

```
This is not a leap year!
```

#### Output 3

```
Insert any year to check for leap year: 2024
```

```
This is a leap year!
```

#### Output 4

```
Insert any year to check for leap year: 2030
```

```
This is not a leap year!
```

### Practice 12: A Program to Check for Existence of a Circle

```
import math

x = float(input("Insert point x: "))
y = float(input("Insert point y: "))
r = float(input("Insert the radius: "))

hypotenuse = math.sqrt(pow(x,2) + pow(y,2))

if hypotenuse <= r:
    print("The point belongs to circle.")
else:
    print("The point does not belong to circle.")
```

#### Output 1

```
Insert point x: 5
Insert point y: 4
Insert the radius: 1

The point does not belong to circle.
```

#### Output 2

```
Insert point x: 3
Insert point y: 3
Insert the radius: 6

The point belongs to circle.
```



### Practice 13: A Program to create quadratic Equation

```
from math import sqrt

x = float(input("Insert x = "))
y = float(input("Insert y = "))
z = float(input("Insert z = "))

a = y*y-4*x*z

if a>0:
    x1 = (-y + sqrt(a))/(2*x)
    x2 = (-y - sqrt(a))/(2*x)
    print("x1 = %.2f; x2 = %.2f" % (x1,x2))
elif a==0:
    x1 = -y/(2*x)
    print("x1 = %.2f" % x1)
else:
    print("No roots exist")
```

#### Output 1

```
Insert x = 4
Insert y = 5
Insert z = 6

No roots exist
```

#### Output 2

```
Insert x = 3
Insert y = 45
Insert z = 2

x1 = -0.04; x2 = -14.96
```

### Practice 14: A Program to Make Guess of Random Number

```
from random import randint

guess_num = randint(1,100)
user_input = 0
trial_period = 1

while guess_num != user_input:
    print("The trial no %d: " % trial_period, end="")
    user_input = int(input())
    if user_input < guess_num:
        print("The input number is less")
    elif user_input > guess_num:
        print("The input number is much")
    else:
        print("Kudos! The right guess...")
    trial_period += 1
```

#### Output 1

```
The trial no 1: 56
The input number is less
The trial no 2: 1000
The input number is much
The trial no 3: 50
The input number is less
The trial no 4: 100
Kudos! The right guess...
```

#### Output 2

```
The trial no 1: 200
The input number is much
The trial no 2: 25
The input number is much
The trial no 3: 15
Kudos! The right guess...
```

### Practice 15: A Program to Print Out the ASCII Table

```
for i in range(32, 128):
    print(chr(i), end=' ')
    if(i-1)%10==0:
        print()
print()
```

#### Output

```
! " # $ % & ' ( )
* + , - . / 0 1 2 3
4 5 6 7 8 9 : ; < =
> ? @ A B C D E F G
H I J K L M N O P Q
R S T U V W X Y Z [
\ ] ^ _ ` a b c d e
f g h i j k l m n o
p q r s t u v w x y
z { | } ~
```

**Practice 16: A Program to Create a Multiplication Table Using While Loop**

```
x = 1

while x < 10:
    y = 1
    while y < 10:
        print("%4d" % (x*y), end="")
        y += 1
    print()
    x += 1
```

**Output**

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81



**Practice 17: A Program to Create Multiplication Table Using for Loop**

```
for x in range(1,10):  
    for y in range(1,10):  
        print("%4d" % (x*y), end="")  
    print()
```

**Output**

1	2	3	4	5	6	7	8	9
2	4	6	8	10	12	14	16	18
3	6	9	12	15	18	21	24	27
4	8	12	16	20	24	28	32	36
5	10	15	20	25	30	35	40	45
6	12	18	24	30	36	42	48	54
7	14	21	28	35	42	49	56	63
8	16	24	32	40	48	56	64	72
9	18	27	36	45	54	63	72	81

TECHVIDYA

### Practice 18: A Program to Convert from Base 2 to 9

```
num1 = int(input("Insert number to convert: "))
x = num1
base_num = int(input("Choose the base(2-9): "))

if not(2<= base_num <=9):
    quit()

num2 = ""

while num1>0:
    num2 = str(num1%base_num) + num2
    num1 //= base_num
output = "The value of {} in base {} is {}"
print(output.format(x, base_num, num2))
```

#### Output 1

```
Insert number to convert: 31
Choose the base(2-9): 4

The value of 31 in base 4 is 133
```

#### Output 2

```
Insert number to convert: 32
Choose the base(2-9): 2

The value of 32 in base 2 is 100000
```

#### Output 3

```
Insert number to convert: 10
Choose the base(2-9): 2

The value of 10 in base 2 is 1010
```

### Practice 19. A Program to Build a Simple Calculator

```
print("Zero operation terminates program!")
while True:
    o = input("Choose Operator(+, -, *, /): ")
    if o == "0":
        break
    if o in ('+', '-', '*', '/'):
        x = float(input("Enter the value of x = "))
        y = float(input("Enter the value of y = "))
        if o == '+':
            print("%.2f" % (x+y))
        elif o == '-':
            print("%.2f" % (x-y))
        elif o == '*':
            print("%.2f" % (x*y))
        elif o == '/':
            if y != 0:
                print("%.2f" % (x / y))
            else:
                print("Error! Division by zero...")
        else:
            print("Invalid operator")
```

#### Output 1

```
Zero operation terminates program!
Choose Opearator(+, -, *, /): +
Enter the value of x = 5
Enter the value of y = 8
13.00
```

#### Output 2

```
Zero operation terminates program!
Choose Opearator(+, -, *, /): *
Enter the value of x = 6
Enter the value of y = 3
18.00
```

**Practice 20: A Program to Detect Number of Digits in an Integer**

```
x = int(input("Insert multiple integer numbers: "))
x = abs(x)

count = 1;
x //= 10

while x > 0:
    x//=10
    count += 1
print("The number of integer(s) is: ",count)
```

**Output 1**

```
Insert multiple integer numbers: 123455432
The number of integer(s) is:  9
```

**Output 2**

```
Insert multiple integer numbers: 6893
The number of integer(s) is:  4
```

**Output 3**

```
Insert multiple integer numbers: 764321
The number of integer(s) is:  6
```

### Practice 21: A Program to Get Sum and Products of Digits

```
str1 = str(input("Insert values only: "))
```

```
sum_digit = 0
```

```
pro_digit = 1
```

```
for x in str1:
```

```
    if x.isdigit() == True:
```

```
        z = int(x)
```

```
        sum_digit = sum_digit + z
```

```
        pro_digit *= z
```

```
print("Sum of digits = ",sum_digit)
```

```
print("Product of digits = ", pro_digit)
```

#### Output 1

```
Insert values only: 1234
```

```
Sum of digits = 10
```

```
Product of digits = 24
```

#### Output 2

```
Insert values only: 5678
```

```
Sum of digits = 26
```

```
Product of digits = 1680
```

#### Output 3

```
Insert values only: 23rt67j9
```

```
Sum of digits = 27
```

```
Product of digits = 2268
```



## Practice 22: A Program to Make a Binary Search of Number in an Array

```

from random import random
N = 20
array = []
for x in range(N):
    array.append(int(random()*100))

array.sort()
print(array)

number = int(input("Search for any number in the array: "))

mini = 0
maxi = N-1

while mini <= maxi:
    mid = (mini + maxi) // 2
    if number < array[mid]:
        maxi = mid-1
    elif number > array[mid]:
        mini = mid+1
    else:
        print("The number is located at index: ", mid)
        break
else:
    print("There is no number!")

```

### Output 1

```

[10, 18, 21, 26, 28, 31, 38, 41, 43, 51, 51, 52, 59, 69, 75, 81, 82, 91, 91, 94]
Search for any number in the array: 82
The number is located at index: 16

```

### Output 2

```

[8, 12, 15, 18, 33, 36, 44, 48, 50, 64, 64, 65, 70, 75, 77, 77, 80, 80, 99, 99]
Search for any number in the array: 33
The number is located at index: 4

```

**Practice 23: A Program to Get Sum of N Series of an Element**

#1,-0.5,0.25,-0.125...

```
x = int(input("Insert number of elements in the series: "))
```

```
y = 1
```

```
z = 0
```

```
sum = 0
```

```
while z < x:
```

```
    sum += y
```

```
    y = y/-2
```

```
    z += 1
```

```
print(sum)
```

**Output 1**

```
Insert number of elements in the series: 123
```

```
0.6666666666666667
```

**Output 2**

```
Insert number of elements in the series: 1
```

```
1
```

**Output 3**

```
Insert number of elements in the series: 23
```

```
0.6666667461395264
```

**Practice 24: A Program to Get Number of Even and Odd Digits**

```
x = int(input("Insert some numbers: "))

ev = 0
od = 0

while x > 0:
    if x%2 == 0:
        ev += 1
    else:
        od += 1
    x = x//10
print("Even numbers = %d, Odd numbers = %d" % (ev,od))
```

**Output 1**

```
Insert some numbers: 12345
Even numbers = 2, Odd numbers = 3
```

**Output 2**

```
Insert some numbers: 123456789
Even numbers = 4, Odd numbers = 5
```

### Practice 25: A Program to Get Factorial Using a While Loop

```
x = abs(int(input("Insert any number: ")))

factorial = 1
while x > 1:
    factorial *= x
    x -= 1
print("The result of factorial = ", factorial)
```

#### Output 1

```
Insert any number: 5
The result of factorial = 120
```

#### Output 2

```
Insert any number: 7
The result of factorial = 5040
```

#### Output 3

```
Insert any number: 10
The result of factorial = 3628800
```

#### Output 4

```
Insert any number: 3
The result of factorial = 6
```

#### Output 5

```
Insert any number: 12
The result of factorial = 479001600
```

### Practice 26: A Program to Get Factorial Using for Loop

```
x = abs(int(input("Insert any number: ")))

factorial = 1
for i in range(2, x+1):
    factorial *= i

print("The result of factorial = ",factorial)
```

#### Output 1

```
Insert any number: 5
The result of factorial = 120
```

#### Output 2

```
Insert any number: 7
The result of factorial = 5040
```

#### Output 3

```
Insert any number: 8
The result of factorial = 40320
```

#### Output 4

```
Insert any number: 4
The result of factorial = 24
```



### Practice 27: A Program to Create a Fibonacci Sequence

```
x = abs(int(input("Insert range of sequence: ")))
```

```
f1 = f2 = 1
```

```
print(f1,f2,end=" ")
```

```
for y in range(x-2):
```

```
    print(f1+f2, end=" ")
```

```
    f1, f2 = f2, f1+f2
```

#### Output 1

```
Insert range of sequence: 6
```

```
1 1 2 3 5 8
```

#### Output 2

```
Insert range of sequence: 10
```

```
1 1 2 3 5 8 13 21 34 55
```

#### Output 3

```
Insert range of sequence: 12
```

```
1 1 2 3 5 8 13 21 34 55 89 144
```

#### Output 4

```
Insert range of sequence: 20
```

```
1 1 2 3 5 8 13 21 34 55 89 144 233 377 610 987 1597 2584 4181 6765
```

**Practice 28: A Program to Get the Value of Fibonacci Element**

```
x = abs(int(input("Choose the element to find its value: ")))  
  
f1 = f2 = 1  
y = 2  
  
while y < x:  
    f1, f2 = f2, f1+f2  
    y += 1  
print("The value of the ",x," element is =", f2)
```

**Output 1**

```
Choose the element to find its value: 6  
The value of the 6 element is = 8
```

**Output 2**

```
Choose the element to find its value: 8  
The value of the 8 element is = 21
```

**Output 3**

```
Choose the element to find its value: 7  
The value of the 7 element is = 13
```

**Output 4**

```
Choose the element to find its value: 20  
The value of the 20 element is = 6765
```

**Practice 29: A Program to Get Find the Greatest Common Divisor**

```
x = int(input("Insert first number: "))
y = int(input("Insert second number: "))

while x != 0 and y != 0:
    if x > y:
        x %= y
    else:
        y &= x

GCD = x + y
print("The greatest common divisor = ",GCD)
```

**Output 1**

```
Insert first number: 50
Insert second number: 40
The greatest common divisor =  2
```

**Output 2**

```
Insert first number: 10
Insert second number: 20
The greatest common divisor =  10
```

**Output 3**

```
Insert first number: 10
Insert second number: 8
The greatest common divisor =  2
```

**Practice 30: A Program to Get Maximum Value of a Floating-Point Number**

```
x = float(input("Insert floating point numbers only: "))

y = str(x)
maxi = -1

for i in range(len(y)):
    if y[i] == '.':
        continue
    elif maxi < int(y[i]):
        maxi = int(y[i])

print("The maximum element is = ",maxi)
```

**Output 1**

```
Insert floating point numbers only: 4.9
The maximum element is = 9
```

**Output 2**

```
Insert floating point numbers only: 3.7
The maximum element is = 7
```

**Output 3**

```
Insert floating point numbers only: 5.2
The maximum element is = 5
```

### Practice 31: A Program to Detect Prime Numbers

```
import math

x = int(input("Insert any number to check: "))

if x < 2:
    print("Please, insert number greater than or equal to 2")
    quit()
elif x == 2:
    print("This is a prime number!")
    quit()

y = 2
num = int(math.sqrt(x))

while y <= num:
    if x % y == 0:
        print("This is NOT a prime number")
        quit()
    y += 1

print("This is a prime number")
```

#### Output 1

```
Insert any number to check: 15
This is NOT a prime number
```

#### Output 2

```
Insert any number to check: 19
This is a prime number
```



### Practice 32: A Program for Quadratic Equations with Solutions at Specified Range of Coefficient

```
import math
x1 = int(input("Insert the value of x1: "))
x2 = int(input("Insert the value of x2: "))
y1 = int(input("Insert the value of y1: "))
y2 = int(input("Insert the value of y2: "))
z1 = int(input("Insert the value of z1: "))
z2 = int(input("Insert the value of z2: "))

x = range(x1, x2+1)
y = range(y1, y2+1)
z = range(z1, z2+1)

for i in x:
    if i == 0:
        continue
    for j in y:
        for k in z:
            print(i, j, k, end=' ')
            A = j*j-4*i*k
            if A >= 0:
                x1 = (-j - math.sqrt(A))/(2*i)
                x2 = (-j + math.sqrt(A))/(2*i)
                print("Valid", round(x1,2), round(x2,2))
            else:
                print("Invalid")
```

#### Output 1

```
Insert the value of x1: -1
Insert the value of x2: 1
Insert the value of y1: 0
Insert the value of y2: 1
Insert the value of z1: -1
Insert the value of z2: 1
-1 0 -1 Invalid
```

```
-1 0 0 Valid -0.0 -0.0
-1 0 1 Valid 1.0 -1.0
-1 1 -1 Invalid
-1 1 0 Valid 1.0 -0.0
-1 1 1 Valid 1.62 -0.62
1 0 -1 Valid -1.0 1.0
1 0 0 Valid 0.0 0.0
1 0 1 Invalid
1 1 -1 Valid -1.62 0.62
1 1 0 Valid -1.0 0.0
1 1 1 Invalid
```

## Output 2

```
Insert the value of x1: -1
Insert the value of x2: 1
Insert the value of y1: 1
Insert the value of y2: 1
Insert the value of z1: -1
Insert the value of z2: 1
-1 1 -1 Invalid
-1 1 0 Valid 1.0 -0.0
-1 1 1 Valid 1.62 -0.62
1 1 -1 Valid -1.62 0.62
1 1 0 Valid -1.0 0.0
1 1 1 Invalid
```

### Practice 33: A Program to Reverse Numbers

```
x = int(input("Insert some numbers: "))
y = 0
z = x
while x != 0:
    digit = x%10
    x = x//10
    y = y*10
    y = y+digit
print("The reversed of ",z," = ",y)
```

#### Output 1

```
Insert some numbers: 78463
The reversed of  78463  =  36487
```

#### Output 2

```
Insert some numbers: 1234
The reversed of  1234  =  4321
```

#### Output 3

```
Insert some numbers: 6789
The reversed of  6789  =  9876
```

### Practice 34: A Program to Expand Strings of Alphabet

```
str1 = input("Insert starting letter: ")
str2 = input("Insert ending letter: ")

while str1 <= str2:
    print(str1, end=" ")
    str1 = chr(ord(str1) + 1)
print()
```

#### Output 1

```
Insert starting letter: a
Insert ending letter: z
a b c d e f g h i j k l m n o p q r s t u v w x y z
```

#### Output 2

```
Insert starting letter: A
Insert ending letter: Z
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z
```

#### Output 3

```
Insert starting letter: h
Insert ending letter: p
h i j k l m n o p
```

#### Output 4

```
Insert starting letter: J
Insert ending letter: Q
J K L M N O P Q
```

### Practice 35: A Program to Replace a Substring of a String

```
str = "Hello, World, Table, Chair, Cup, Tea"
print(str)

substr1 = input("Choose an old substring to replace: ")
substr2 = input("Insert new substring: ")
lensubstr1 = len(substr1)

while str.find(substr1) > 0:
    i = str.find(substr1)
    str = str[:i] + substr2 + str[i+lensubstr1:]
print(str)
```

#### Output 1

```
Hello, World, Table, Chair, Cup, Tea
Choose an old substring to replace: World
Insert new substring: Python
Hello, Python, Table, Chair, Cup, Tea
```

#### Output 2

```
Hello, World, Table, Chair, Cup, Tea
Choose an old substring to replace: Tea
Insert new substring: Python
Hello, World, Table, Chair, Cup, Python
```



**Practice 36. A Program to Select Integers from String**

```
str = input("Insert strings with integer values: ")
lenstr = len(str)
mynum = []
x = 0
while x < lenstr:
    num = ""
    symbol = str[x]
    while '0' <= symbol <= '9':
        num += symbol
        x += 1
    if x < lenstr:
        symbol = str[x]
    else:
        break
    x += 1
    if num != "":
        mynum.append(int(num))
print(mynum)
```

**Output 1**

Insert strings with integer values: 3k6g1w23kt98sdj2kuy8

[3, 6, 1, 23, 98, 2, 8]

### Practice 37: A Program to Sort Words According to Their Length

```
str = input("Insert different strings: ")
first = str.split()
len_first = len(first)

for i in range(len_first - 1):
    for j in range(len_first - 1 - i):
        if len(first[j]) > len(first[j + 1]):
            first[j], first[j + 1] = first[j + 1], first[j]
print(' '.join(first))
```

#### Output 1

Insert different strings: School, Academy, Tea, At, Hello, World,  
At, Tea, World, Hello, School, Academy

#### Output 2

Insert different strings: Principal, College, Root, C++, C#,  
C#, C++, Root, College, Principal,

#### Output 3

Insert different strings: Master Tell Manchester United Chelsea King  
Tell King Master United Chelsea Manchester

### Practice 38: A Program to Find the Longest Word in a String

```
str = input("Insert some strings: ")
totalstr = str.split()

longest = 0

for i in range(1, len(totalstr)):
    if len(totalstr[longest]) < len(totalstr[i]):
        longest = i
print(totalstr[longest])
```

#### Output 1

```
Insert some strings: Table King Manchester College Top
Manchester
```

#### Output 2

```
Insert some strings: Hello Christmas General Queen
Christmas
```

#### Output 3

```
Insert some strings: Jaquar LG Hiesense Vehicle kettle
Hiesense
```

**Practice 39. A Program to Get Percentage of Uppercase and Lowercase**

```
str = input("Insert some strings of Uppercase and Lowercase: ")
```

```
len_str = len(str)
```

```
upper = lower = 0
```

```
for i in str:
```

```
    if 'a' <= i <= 'z':
```

```
        lower += 1
```

```
    elif 'A' <= i <= 'Z':
```

```
        upper += 1
```

```
print("Percentage of Uppercase: %.2f %% " % (upper/len_str * 100))
```

```
print("Percentage of Lowercase: %.2f %% " % (lower/len_str * 100))
```

**Output 1**

```
Insert some strings of Uppercase and Lowercase: GhtHkUTwerX
```

```
Percentage of Uppercase: 45.45 %
```

```
Percentage of Lowercase: 54.55 %
```

**Output 2**

```
Insert some strings of Uppercase and Lowercase: HtuDESresBZAqre
```

```
Percentage of Uppercase: 46.67 %
```

```
Percentage of Lowercase: 53.33 %
```

### Practice 40: A Program to Check for String Palindrome

```
str = input("Insert a string: ")
len_str = len(str)

for i in range(len_str//2):
    if str[i] != str[-1-i]:
        print("This is NOT a palindrome!")
        quit()
print("This is a PALINDROME!")
```

#### Output 1

```
Insert a string: HelloWorld
This is NOT a palindrome!
```

#### Output 2

```
Insert a string: HellolleH
This is a PALINDROME!
```

#### Output 3

```
Insert a string: Hellolleh
This is NOT a palindrome!
```



**Practice 41: A Program to Generate Random Numbers Using Arrays**

```
from random import randint

x = 20
array = []

for i in range(x):
    array.append(randint(1,10))
for i in array:
    print(i,end=' ')
print()
```

**Output 1**

7 8 9 9 6 9 4 8 8 4 10 5 6 5 5 10 3 9 3 1

**Output 2**

10 4 3 2 7 9 2 10 7 8 4 7 8 8 7 4 10 2 5 7

**Output 3**

10 4 4 8 7 1 10 1 7 8 8 2 6 9 5 6 4 2 2 2

**Output 4**

2 9 10 9 1 9 8 5 3 6 3 10 6 5 2 9 8 4 2 9

### Practice 42: A Program to Get the Maximum Element in an Array

```
from random import random
x = 20
y = []
for i in range(x):
    y.append(random())
    print(round(y[i],2), end=' ')
print()
```

```
maximum = 0
for i in y:
    if i > maximum:
        maximum = i
print("The maximum value = ", round(maximum,2))
```

#### Output 1

```
0.01 0.61 0.47 0.99 0.47 0.79 0.97 0.24 0.35 0.27 0.9 0.45 0.35 0.07 0.38 0.94 0.55
0.06 0.57 0.86
```

The maximum value = 0.99

#### Output 2

```
0.75 0.27 0.51 0.36 0.08 0.17 0.95 0.97 0.37 0.59 0.0 0.51 0.4 0.85 0.33 0.77 0.83
0.82 0.09 0.54
```

The maximum value = 0.97

### Practice 43: A Program to Get the Minimum Element in an Array

```
from random import random
x = 20
y = []
for i in range(x):
    y.append(random())
    print(round(y[i],2), end=' ')
print()
```

```
minimum = 1

for i in y:
    if i < minimum:
        minimum = i
print("The minimum element is: ", round(minimum, 2))
```

#### Output 1

```
0.57 0.25 0.71 0.48 0.01 0.04 0.22 0.6 0.42 0.91 0.87 0.03 0.73 0.57 0.97 0.67 0.03
0.35 0.06 0.85
The minimum element is: 0.01
```

#### Output 2

```
0.83 0.23 0.92 0.02 0.49 0.35 0.44 0.52 0.94 0.19 0.97 0.75 0.89 0.15 0.35 0.64
0.85 0.67 0.81 0.04
The minimum element is: 0.02
```

**Practice 44: A Program to Get the Number of Even and Odd Numbers**

```
import random
x = []
for i in range(10):
    x.append(int(random.random()*1000))
print(x)
even = odd = 0
for i in x:
    if i%2 == 0:
        even += 1
    else:
        odd += 1
print("The number of even = ", even)
print("The number of odd = ", odd)
```

**Output 1**

```
[205, 891, 152, 260, 750, 866, 744, 839, 128, 655]
```

```
The number of even = 6
```

```
The number of odd = 4
```

**Output 2**

```
[521, 125, 530, 605, 39, 831, 567, 455, 440, 223]
```

```
The number of even = 2
```

```
The number of odd = 8
```

**Practice 45. A Program to Get Positive Numbers Out of Negative Numbers**

```

import random
x = []
for i in range(20):
    x.append(int(random.random()*20)-10)
print(x)
pos = []
neg = []
for i in x:
    if i<0:
        neg.append(i)
    elif i>0:
        pos.append(i)
print("Negative numbers = ",neg)
print("Positive numbers = ",pos)

```

**Output 1**

```
[6, 7, -5, -8, 3, -7, -5, 6, 4, 7, 9, -3, 2, -7, -4, -6, -7, 1, -4, 0]
```

```
Negative numbers = [-5, -8, -7, -5, -3, -7, -4, -6, -7, -4]
```

```
Positive numbers = [6, 7, 3, 6, 4, 7, 9, 2, 1]
```

**Output 2**

```
[9, 6, -9, -6, -6, 5, -7, -8, 3, -9, -6, 4, -3, 6, 2, -2, -8, 5, 5, 8]
```

```
Negative numbers = [-9, -6, -6, -7, -8, -9, -6, -3, -2, -8]
```

```
Positive numbers = [9, 6, 5, 3, 4, 6, 2, 5, 5, 8]
```



### Practice 46: A Program to Get Numbers Greater than the Average of an Array

```
from random import random
x = 10
y = []
avg = 0
for i in range(x):
    y.append(random())
    print("%5.2f" % y[i], end="")
    avg += y[i]
print()
average = avg/x
print("The average of the array = %.2f" % average)
print("Numbers greater than the average number are below: ")
for i in y:
    if i > average:
        print("%4.2f" % i)
```

#### Output 1

```
0.61 0.18 0.59 0.82 0.74 0.50 0.22 0.61 0.24 0.03
The average of the array = 0.45
Numbers greater than the average number are below:
0.61
0.59
0.82
0.74
0.50
0.61
```

**Practice 47: A Program to Replace List Items With -1, 0, 1**

```
listitem = [6,8,9,-2,0,1,-3,5,8,0,-6]  
print(listitem)
```

```
for i in range(len(listitem)):  
    if listitem[i] > 0:  
        listitem[i] = 1  
    elif listitem[i] < 0:  
        listitem[i] = -1  
    else:  
        listitem[i] = 0
```

```
print(listitem)
```

**Output**

```
[6, 8, 9, -2, 0, 1, -3, 5, 8, 0, -6]  
[1, 1, 1, -1, 0, 1, -1, 1, 1, 0, -1]
```

# TECHVIDYA

### Practice 48: A Program to Check for File Extension

```
exe = ['gif','png','jpeg','jpg','svg','txt']

filexe = input("Insert file with extension: ").split('.')
if len(filexe) >= 2:
    Extension = filexe[-1].lower()
    if Extension in exe:
        print("File extension exist")
    else:
        print("File extension does not exist")
else:
    print("File does not have extension")
```

#### Output 1

```
Insert file with extension: C:\Users\USER\Pictures\EloGod Fish Stock
File does not have extension
```

#### Output 2

```
Insert file with extension: C:\Users\USER\Pictures\EloGod.png
File extension exist
```

#### Output 3

```
Insert file with extension: money.jpeg
File extension exist
```

**Practice 49: A Program to Remove Exclamation Mark (!) from Text**

```
str = input("Insert some text with exclamation mark(!): \n")
symbols = ['!']
```

```
listitem = str.split()
```

```
x = 0
for i in listitem:
    if i[-1] in symbols:
        listitem[x] = i[:-1]
        i = listitem[x]
    if i[0] in symbols:
        listitem[x] = i[1:]
    x += 1
```

```
x = 0
while x < len(listitem):
    print(listitem[x], end=' ')
    x += 1
    if x%5 == 0:
        print()
```

**Output 1**

```
Insert some text with punctuation mark(!):
Helloworld!
Helloworld
```

**Output 2**

```
Insert some text with punctuation mark(!):
Welcome!
Welcome
```

**Practice 50: A Program to get Intersection of List Using for Loop**

```
x = [6,8,9,10,[3,6,4],'t','tt','u',5]
```

```
y = [8,19,6,[3,6,4],'tt','u','z']
```

```
z = []
```

```
for i in x:
```

```
    for j in y:
```

```
        if i == j:
```

```
            z.append(i)
```

```
            break
```

```
print(z)
```

**Output**

```
[6, 8, [3, 6, 4], 'tt', 'u']
```

TECHVIDYA



**Practice 51: A Program for Simple Intersection of List**

```
x = [6,8,9,10,'t','tt','u',5]
```

```
y = [8,19,6,'tt','u','z']
```

```
z = list(set(x) & set(y))
```

```
print(z)
```

**Output**

```
[8, 'u', 'tt', 6]
```



### Practice 52: A Program for Longest ordered Sequence in Ascending Order

```

from random import random

num = 20
listitem = [0]*num

for i in range(num):
    listitem[i] = int(random()*50)
print(listitem)

maxi = 1
mylength = 1
mycode = 0

for i in range(1,num):
    if listitem[i] > listitem[i-1]:
        mylength += 1
    else:
        if mylength > maxi:
            maxi = mylength
            mycode = i
        mylength = 1
print("The maximum length = ",maxi)
print("The ordered values are = ",listitem[mycode-maxi : mycode])

```

#### Output 1

```

[43, 27, 28, 40, 43, 23, 29, 47, 37, 2, 14, 14, 18, 4, 27, 36, 24, 31, 42, 29]
The maximum length = 4
The ordered values are = [27, 28, 40, 43]

```

#### Output 2

```

[21, 37, 16, 16, 37, 20, 21, 3, 3, 10, 15, 40, 44, 40, 22, 0, 40, 26, 11, 6]
The maximum length = 5
The ordered values are = [3, 10, 15, 40, 44]

```

### Practice 53: A Program to Get the Most Occurrent Element

```
from random import random

x = [int(random()*100)for i in range(20)]
print(x)

myset = set(x)

highest = None
frequent = 0

for item in myset:
    freq = x.count(item)

    if freq > frequent:
        frequent = freq
        highest = item
print("The highest occurrent element is ", highest)
```

#### Output 1

```
[40, 7, 81, 71, 20, 3, 55, 44, 12, 81, 62, 22, 53, 40, 21, 74, 68, 58, 52, 28]
The highest occurrent element is 40
```

#### Output 2

```
[78, 46, 75, 67, 26, 70, 52, 9, 23, 9, 13, 96, 34, 2, 84, 82, 44, 88, 21, 30]
The highest occurrent element is 9
```

#### Output 3

```
[96, 88, 89, 28, 76, 77, 31, 29, 44, 32, 29, 28, 11, 7, 42, 25, 58, 6, 96, 73]
The highest occurrent element is 96
```

### Practice 54: A Program to Bubble Sort Elements of an Array

```
from random import randint
```

```
x = 20
```

```
y = []
```

```
for i in range(20):
```

```
    y.append(randint(1,20))
```

```
print(y)
```

```
for i in range(x):
```

```
    for j in range(x-i-1):
```

```
        if y[j] > y[j + 1]:
```

```
            z = y[j]
```

```
            y[j] = y[j+1]
```

```
            y[j+1] = z
```

```
print(y)
```

#### Output 1

```
[9, 4, 20, 11, 15, 5, 11, 10, 17, 8, 16, 5, 16, 5, 19, 10, 5, 9, 20, 19]
```

```
[4, 5, 5, 5, 5, 8, 9, 9, 10, 10, 11, 11, 15, 16, 16, 17, 19, 19, 20, 20]
```

#### Output 2

```
[14, 4, 7, 16, 18, 14, 20, 18, 13, 12, 11, 13, 3, 10, 8, 20, 8, 13, 16, 18]
```

```
[3, 4, 7, 8, 8, 10, 11, 12, 13, 13, 13, 14, 14, 16, 16, 18, 18, 18, 20, 20]
```

#### Output 3

```
[6, 14, 12, 4, 2, 3, 3, 8, 11, 19, 7, 5, 9, 9, 19, 14, 12, 3, 7, 16]
```

```
[2, 3, 3, 3, 4, 5, 6, 7, 7, 8, 9, 9, 11, 12, 12, 14, 14, 16, 19, 19]
```

### Practice 55: A Program to Sort Array Using Selection Sorting

```
from random import randint
```

```
x = 20
```

```
y = []
```

```
for i in range(20):
```

```
    y.append(randint(1,20))
```

```
print(y)
```

```
j = x-1
```

```
while j != 0:
```

```
    k = 0
```

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

```
        if y[i] > y[k]:
```

```
            k = i
```

```
    z = y[k]
```

```
    y[k] = y[j]
```

```
    y[j] = z
```

```
    j -= 1
```

```
print(y)
```

#### Output 1

```
[2, 14, 11, 5, 20, 13, 9, 20, 15, 9, 17, 8, 1, 14, 12, 20, 10, 16, 12, 14]
```

```
[1, 2, 5, 8, 9, 9, 10, 11, 12, 12, 13, 14, 14, 14, 15, 16, 17, 20, 20, 20]
```

#### Output 2

```
[5, 8, 19, 13, 19, 3, 19, 14, 5, 14, 11, 15, 20, 17, 2, 11, 9, 20, 1, 12]
```

```
[1, 2, 3, 5, 5, 8, 9, 11, 11, 12, 13, 14, 14, 15, 17, 19, 19, 19, 20, 20]
```

#### Output 3

```
[11, 2, 6, 14, 20, 6, 8, 10, 13, 14, 6, 5, 14, 7, 12, 3, 15, 2, 11, 15]
```

```
[2, 2, 3, 5, 6, 6, 6, 7, 8, 10, 11, 11, 12, 13, 14, 14, 14, 15, 15, 20]
```



### Practice 56: A Program to Generate Matrix of Random Numbers

```
from random import randint

row = 6
col = 6
x = []

for i in range(row):
    y = []
    for j in range(col):
        y.append(randint(1,100))
    x.append(y)

for i in x:
    for j in i:
        print("%3d" % j, end=' ')
    print()
```

#### Output 1

```
81 24 89 25 41 9
45 75 64 18 87 39
82 82 52 39 84 80
2 96 82 82 26 27
35 48 90 95 93 37
96 83 44 43 56 24
```

#### Output 2

```
24 6 70 70 27 64
47 66 68 2 56 15
45 78 45 38 26 73
81 44 22 36 98 6
32 22 44 27 86 53
76 61 70 19 53 81
```

### Practice 57: A Program to Get the Rows and Columns with Maximum Sum of Elements

```
from random import random

matrix = []

for i in range(6):
    row = []
    for j in range(6):
        row.append(int(random()*10))
    matrix.append(row)
```

```
for row in matrix:
    print(row)
```

```
rmaxi = 0
rid = 0
i = 0
```

```
for row in matrix:
    if sum(row) > rmaxi:
        rmaxi = sum(row)
        rid = i
    i += 1
print('Row ID:',rid, '=', rmaxi)
```

```
cmaxi = 0
cid = 0
for i in range(6):
    sumcol = 0
    for j in range(6):
        sumcol += matrix[j][i]
    if sumcol > cmaxi:
        cmaxi = sumcol
        cid = i
print('Column ID:',cid, '=', cmaxi)
```

**Output 1**

```
[1, 9, 5, 6, 8, 1]
[9, 2, 0, 4, 2, 7]
[8, 5, 6, 3, 5, 8]
[8, 4, 5, 4, 6, 7]
[6, 8, 3, 6, 0, 1]
[4, 0, 8, 4, 4, 1]
Row ID: 2 = 35
Column ID: 0 = 36
```

**Output 2**

```
[9, 8, 1, 9, 1, 8]
[5, 2, 2, 3, 6, 0]
[3, 2, 1, 8, 1, 4]
[3, 6, 5, 2, 1, 6]
[4, 6, 6, 7, 1, 0]
[3, 5, 0, 1, 6, 3]
Row ID: 0 = 36
Column ID: 3 = 30
```

**Output 3**

```
[7, 9, 5, 7, 0, 2]
[2, 4, 4, 1, 5, 5]
[9, 1, 3, 9, 8, 2]
[2, 0, 7, 0, 7, 4]
[0, 3, 6, 8, 8, 4]
[2, 8, 0, 8, 1, 9]
Row ID: 2 = 32
Column ID: 3 = 33
```

### Practice 58: A Program to Sum Items in Rows and Columns of Elements

```
from random import randint

col = 6
row = 6
matrix = []

sum_col = [0]*col
sum_row = [0]*row

for i in range(row):
    myrow = []
    for j in range(col):
        myrow.append(randint(0,3))
    matrix.append(myrow)

for i in range(row):
    for j in range(col):
        sum_row[i] += matrix[i][j]
        sum_col[j] += matrix[i][j]

for i in range(row):
    print(matrix[i], "|", sum_row[i])

print("_" * col * 4)
print(sum_col)
```

#### Output 1

```
[1, 2, 2, 3, 2, 1] | 11
[2, 0, 3, 3, 2, 1] | 11
[0, 2, 3, 1, 1, 2] | 9
[0, 3, 1, 2, 3, 1] | 10
[3, 0, 3, 2, 3, 0] | 11
[2, 2, 1, 2, 1, 2] | 10

[8, 9, 13, 13, 12, 7]
```

### Practice 59: A Program to Sum Diagonals of a Matrix

```

from random import random
row = 5
matrix = []

for i in range(row):
    myrow = []
    for j in range(row):
        myrow.append(int(random()*10))
    matrix.append(myrow)

for myrow in matrix:
    print(myrow)

sum_diagonal1 = 0
sum_diagonal2 = 0

for i in range(row):
    sum_diagonal1 += matrix[i][i]
    sum_diagonal2 += matrix[i][row-i-1]

print('Sum of first diagonal:',sum_diagonal1)
print('Sum of second diagonal:',sum_diagonal2)

```

#### Output

```

[7, 3, 2, 4, 5]
[7, 5, 6, 7, 9]
[1, 1, 9, 6, 4]
[9, 5, 6, 2, 8]
[7, 3, 9, 6, 3]
Sum of first diagonal: 26
Sum of second diagonal: 33

```



### Practice 60: A Program to Interchange the Principal Diagonals of Matrix

```

from random import randint
row = 5
matrix = []
for i in range(row):
    myrow = []
    for j in range(row):
        myrow.append(randint(1,100))
        print("%4d" % myrow[j], end=' ')
    matrix.append(myrow)
    print()
print()
print("Interchanged principal diagonals of matrix below")
for i in range(row):
    x = matrix[i][i]
    matrix[i][i] = matrix[i][row-1-i]
    matrix[i][row-1-i] = x
for i in matrix:
    for j in i:
        print("%4d" % j, end=' ')
    print()

```

#### Output

61	98	17	70	26
43	31	87	44	76
36	33	6	8	97
74	28	34	23	95
23	85	1	67	65

#### Interchanged principal diagonals of matrix below

26	98	17	70	61
43	44	87	31	76
36	33	6	8	97
74	23	34	28	95
65	85	1	67	23

**Practice 61: A Program to Sort Columns of Element by Sorting the First Row**

```

from random import randint
col = 6
row = 3
matrix = []
for i in range(row):
    myrow = []
    for j in range(col):
        myrow.append(randint(10,100))
    matrix.append(myrow)
for i in matrix:
    print(i)
print()
print('Result of sorted column of elements below')
k = col-1
while k != 0:
    z = 0
    for j in range(1, k+1):
        if matrix[0][j] > matrix[0][z]:
            z = j
    for i in range(row):
        y = matrix[i][z]
        matrix[i][z] = matrix[i][k]
        matrix[i][k] = y
    k -= 1
for i in matrix:
    print(i)

```

**Output**

```
[73, 58, 71, 53, 69, 22]
```

```
[52, 89, 52, 85, 90, 73]
```

```
[47, 53, 64, 11, 35, 25]
```

**Result of sorted column of elements below**

```
[22, 53, 58, 69, 71, 73]
```

```
[73, 85, 89, 90, 52, 52]
```

```
[25, 11, 53, 35, 64, 47]
```

### Practice 62: A Program to Check Rows and Columns that Has Particular Element

```
from random import random

row = 5
col = 10
matrix = []
for i in range(row):
    myrow = []
    for j in range(col):
        myrow.append(int(random()*50)+10)
    matrix.append(myrow)

for myrow in matrix:
    print(myrow)

num = int(input("Range of numbers(10-50): "))

print("Rows: ", end=' ')
for i in range(row):
    if num in matrix[i]:
        print(i, end=' ')
print()

print("Columns: ", end=' ')
for j in range(col):
    for i in range(row):
        if matrix[i][j] == num:
            print(j, end=' ')
            break
print()
```

**Output 1**

```
[22, 26, 21, 14, 20, 52, 48, 20, 13, 42]
[30, 16, 42, 29, 42, 27, 12, 34, 41, 16]
[11, 20, 38, 50, 27, 39, 22, 57, 52, 50]
[49, 50, 53, 51, 47, 11, 26, 18, 41, 57]
[31, 48, 26, 30, 39, 11, 37, 41, 28, 13]
Range of numbers(10-50): 22
Rows: 0 2
Columns: 0 6
```

**Output 2**

```
[35, 24, 22, 45, 39, 14, 29, 37, 18, 47]
[54, 31, 17, 44, 22, 33, 57, 19, 28, 53]
[54, 46, 21, 14, 46, 44, 12, 25, 51, 59]
[59, 17, 39, 28, 26, 10, 24, 10, 21, 37]
[17, 53, 28, 14, 10, 51, 57, 29, 10, 12]
Range of numbers(10-50): 51
Rows: 2 4
Columns: 5 8
```

**Output 3**

```
[44, 51, 35, 23, 12, 30, 52, 55, 32, 23]
[40, 49, 19, 39, 42, 17, 30, 52, 36, 24]
[13, 41, 14, 46, 25, 37, 19, 42, 17, 14]
[25, 44, 45, 57, 57, 36, 34, 40, 13, 56]
[31, 45, 24, 34, 39, 46, 47, 19, 55, 42]
Range of numbers(10-50): 34
Rows: 3 4
Columns: 3 6
```



### Practice 63: A Program to Generate Beautiful Unicode

```
Unicode = {0: 9471, 1: 10102, 2: 10103, 3: 10104,
           4: 10105, 5: 10106, 6: 10107, 7: 10108,
           8: 10109, 9: 10110, 10: 10111}
```

```
x = input("Insert digits 0-9: ")
num = " "
```

```
for i in x:
    i = int(i)
    i = chr(Unicode[i])
    num = num + i
```

```
print("The result of Unicode = %s" % num)
```

#### Output 1

Insert digits 0-9: 0123456789

The result of Unicode = 0 1 2 3 4 5 6 7 8 9

#### Output 2

Insert digits 0-9: 678234

The result of Unicode = 6 7 8 2 3 4

#### Output 3

Insert digits 0-9: 08255

The result of Unicode = 0 8 2 5 5



### Practice 64: A Program to Get Prices of Products

```
products = {"Grape":5.9, "Guava": 4.5,
            "Mango":4.8, "Cashew":2.4,
            "Banana":3.0, "Pear": 5.8}

for pro, price in products.items():
    print(pro, " = ", price)
cost = 0
while True:
    pro = input("Select product (n=nothing): ")
    if pro == 'n':
        break
    qty = int(input("Number of product? "))
    cost += products[pro]*qty

print("Price of product(s): ",cost)
```

#### Output 1

```
Grape = 5.9
Guava = 4.5
Mango = 4.8
Cashew = 2.4
Banana = 3.0
Pear = 5.8
Select product (n=nothing): Mango
Number of product? 5
Select product (n=nothing): Pear
Number of product? 2
Select product (n=nothing): n
Price of product(s): 35.6
```

**Practice 65: A Program to Make List of Dictionaries Using 2 Lists**

```
x = ['a','b','c','d','e']  
y = [1,2,3,4,5]  
  
z = {}  
  
for i in range(len(x)):  
    # z[y[i]] = x[i]  
    z = dict(zip(y,x))  
  
print(z)
```

**Output**

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```



TECHVIDYA

### Practice 66. A Program to Delete Dictionary Item

```
import random

x = {1: "a", 2: "b", 3: "c", 4: "d", 5: "e"}
print(x)

keys = list(x.keys())
del_keys = random.choice(keys)
del x[del_keys]

print(x)
```

#### Output 1

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
{1: 'a', 2: 'b', 4: 'd', 5: 'e'}
```

#### Output 2

```
{1: 'a', 2: 'b', 3: 'c', 4: 'd', 5: 'e'}
{2: 'b', 3: 'c', 4: 'd', 5: 'e'}
```

**Practice 67: A Program to Return Value of 2 Arguments Using Function**

```
def avg(num1,num2):  
    x = (num1+num2)/2  
    return x  
  
y = int(input("Insert first value: "))  
z = int(input("Insert second value: "))  
  
average = avg(y,z)  
print(round(average,2))
```

**Output 1**

```
Insert first value: 50  
Insert second value: 45  
47.5
```

**Output 2**

```
Insert first value: 75  
Insert second value: 30  
52.5
```

**Output 3**

```
Insert first value: 69  
Insert second value: 12  
40.5
```

### Practice 68: A Program to Fill List

```
from random import randint

def list_fill(first, qyt, mini, maxi):
    for i in range(qyt):
        first.append(randint(mini,maxi))

minimum = int(input("Insert minimum value: "))
maximum = int(input("Insert maximum value: "))
num = int(input("Number of elements: "))
x = []

list_fill(x,num,minimum,maximum)
print(x)
```

#### Output 1

```
Insert minimum value: 12
Insert maximum value: 30
Number of elements: 10
[22, 24, 12, 18, 19, 17, 19, 23, 21, 19]
```

#### Output 2

```
Insert minimum value: 1
Insert maximum value: 50
Number of elements: 20
[6, 38, 3, 24, 31, 41, 1, 37, 40, 44, 13, 49, 14, 39, 49, 11, 2, 25, 7, 49]
```

#### Output 3

```
Insert minimum value: 4
Insert maximum value: 10
Number of elements: 15
[6, 5, 6, 6, 4, 7, 6, 9, 6, 5, 6, 7, 5, 8, 7]
```



### Practice 69: A Program to Get the Arithmetic Mean of a List

```
def avg_list(first):
    last = len(first)
    sum = 0
    for i in first:
        sum += i
    return sum/last

x = input("Insert some integer values: ")
x = x.split()
for i in range(len(x)):
    x[i] = int(x[i])

average = avg_list(x)

print("The result of the average =",round(average,2))
```

#### Output 1

```
Insert some integer values: 3 4 5 7
The result of the average = 4.75
```

#### Output 2

```
Insert some integer values: 8 4 2 7 9 1
The result of the average = 5.17
```

#### Output 3

```
Insert some integer values: 5 3 6 9 2 3
The result of the average = 4.67
```

**Practice 70: A Program to Generate Fibonacci Sequence Using Function**

```
def fibonacci(list_item):  
    f1 = f2 = 1  
    print(f1, f2, end=' ')  
    while list_item > 2:  
        num = f2  
        f2 = f1 + f2  
        f1 = num  
        print(f2, end=' ')  
        list_item -= 1  
    print()  
  
x = int(input("Insert range of Fibonacci sequence: "))  
fibonacci(x)
```

**Output 1**

```
Insert range of Fibonacci sequence: 5  
1 1 2 3 5
```

**Output 2**

```
Insert range of Fibonacci sequence: 12  
1 1 2 3 5 8 13 21 34 55 89 144
```

**Output 3**

```
Insert range of Fibonacci sequence: 10  
1 1 2 3 5 8 13 21 34 55
```

**Practice 71: A Program to Get Fibonacci Value Using Recursion**

```
x = int(input("Choose a Fibonacci number to get its value: "))
```

```
def fibonacci(x):  
    if x == 1 or x == 2:  
        return 1  
    return fibonacci(x-1) + fibonacci(x-2)  
print(fibonacci(x))
```

**Output 1**

```
Choose a Fibonacci number to get its value: 10  
55
```

**Output 2**

```
Choose a Fibonacci number to get its value: 12  
144
```

**Output 3**

```
Choose a Fibonacci number to get its value: 6  
8
```

### Practice 72: A Program to Get Factorial Using Recursion

```
def factorial(x):  
    if x == 1:  
        return x  
    return x * factorial(x-1)  
y = int(input("Insert any number to calculate factorial: "))  
print(factorial(y))
```

#### Output 1

```
Insert any number to calculate factorial: 5  
120
```

#### Output 2

```
Insert any number to calculate factorial: 12  
479001600
```

#### Output 3

```
Insert any number to calculate factorial: 7  
5040
```

### Practice 73: A Program to Get the LCM

```
def LCM(x,y):  
    z = x*y  
    while x != 0 and y != 0:  
        if x > y:  
            x %= y  
        else:  
            y %= x  
    return z//(x+y)  
  
a = int(input("Insert value of a: "))  
b = int(input("Insert value of b: "))  
  
print("The LCM = ", LCM(a,b))
```

#### Output 1

```
Insert value of a: 8  
Insert value of b: 5  
The LCM = 40
```

#### Output 2

```
Insert value of a: 9  
Insert value of b: 7  
The LCM = 63
```

#### Output 3

```
Insert value of a: 15  
Insert value of b: 12  
The LCM = 60
```



### Practice 74: A Program to Reverse Word Sequence

```
def string_rev(str):  
    str = str.split()  
    str.reverse()  
    return ' '.join(str)  
  
print(string_rev(input("Insert some strings: ")))
```

#### Output 1

```
Insert some strings: Hello World  
World Hello
```

#### Output 2

```
Insert some strings: Welcome To Python Programming  
Programming Python To Welcome
```

#### Output 3

```
Insert some strings: A Program to Reverse Word Sequence  
Sequence Word Reverse to Program A
```

### Practice 75: A Program to Search for Binary Numbers

```

from random import randint

def num_search(first,last):
    mid = len(first) //2
    mini = 0
    maxi = len(first) - 1
    while first[mid] != last and mini <= maxi:
        if last > first[mid]:
            mini = mid + 1
        else:
            maxi = mid -1
        mid = (mini + maxi) // 2
    if mini > maxi:
        return None
    else:
        return mid

x = []
for i in range(15):
    x.append(randint(1,20))
x.sort()
print(x)

num = int(input("Insert any number to search the list: "))
print(num,"is found at index",num_search(x,num))

```

#### Output 1

```

[1, 2, 2, 3, 4, 4, 5, 8, 9, 9, 14, 15, 17, 17, 20]
Insert any number to search the list: 14
14 is found at index 10

```

#### Output 2

```

[1, 1, 3, 4, 6, 6, 7, 9, 11, 13, 14, 14, 16, 17, 18]
Insert any number to search the list: 9
9 is found at index 7

```

**Practice 76: A Program to Make a Ring Shift or Recycle Items of a List**

```
def ring_shift(first,next):  
    if next < 0:  
        next = abs(next)  
        for i in range(next):  
            first.append(first.pop(0))  
    else:  
        for i in range(next):  
            first.insert(0,first.pop())
```

```
values = [9,8,7,6,5,4,3,2,1,0]  
print(values)
```

```
ring_shift(values,-2)  
print(values)
```

```
ring_shift(values, 3)  
print(values)
```

```
ring_shift(values, 5)  
print(values)
```

**Output**

```
[9, 8, 7, 6, 5, 4, 3, 2, 1, 0]  
[7, 6, 5, 4, 3, 2, 1, 0, 9, 8]  
[0, 9, 8, 7, 6, 5, 4, 3, 2, 1]  
[5, 4, 3, 2, 1, 0, 9, 8, 7, 6]
```

### Practice 77: A Program to Read Text

```
text = []  
for i in open("C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt"):  
    text.append(i)  
print(text)  
  
for i in range(len(text)):  
    if text[i][-1] == '\n':  
        text[i] = text[i][: -1]  
print(text)
```

#### Output

**Change the file path below to your file location**

C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt

TECHVIDYA

### Practice 78: A Program to Use Read Method

```
fp = open("C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt")
data = fp.read()
fp.close()

print(repr(data))

data = data.split('\n')
print(data)
```

#### Output

**Change the file path below to your file location**

C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt





### Practice 79: A Program to Use ReadLine Method

```
fp = open("C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt")
data = []

i = fp.readline()
while i != "":
    data.append(i)
    i = fp.readline()
fp.close()
print(data)
```

#### Output

**Change the file path below to your file location**

C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt

TECHVIDYA

### Practice 80: A Program to Use ReadLines Method

```
fp = open("C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt")  
data = fp.readlines()  
fp.close()  
  
print(data)
```

#### Output

**Change the file path below to your file location**

C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt



### Practice 81: A Program to Write to File

```
fp = open("C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt",'w')

fp.write("Hello World 1 \n")
fp.write("Hello World 2 \n")

fp.writelines("Hello World 3 \n")
text = ["Hello World 4 \n", "Hello World 5 \n", "Hello World 6"]
fp.writelines(text)

fp.close()

fp = open("C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt")
print(fp.read())
fp.close()
```

#### Output

**Change the file path below to your file location**

**C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt**

### Practice 82: A Program to Read Text from File to Dictionary

```
goods = {}

for i in open("C:/Users/USER/Desktop/PythonProjects/Python_File/Python.txt"):
    cate = i.split()
    cate[1] = float(cate[1])
    cate[2] = int(cate[2])
    goods[cate[0]] = cate[1:]
print(goods)
```

#### Output

**Change the file path below to your file location**

C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt



### Practice 83: A Program to Count Number of Lines, Words and Letters in a Text File

```
lines = 0
words = 0
letters = 0

fp = "C:/Users/USER/Desktop/Python Projects/Python_File/Python.txt"

for line in open(fp):
    lines += 1
    letters += len(line)

    pos = 'out'
    for letter in line:
        if letter != ' ' and pos == 'out':
            words += 1
            pos = 'in'
        elif letter == ' ':
            pos = 'out'
print("Lines: ",lines)
print("Words: ",words)
print("Letters: ",letters)
```

#### Output

**Change the file path below to your file location**

C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt



### Practice 84: A Program to Capture String Errors

```
x = input("Insert numbers only: ")

while type(x) != float:
    try:
        x = float(x)
    except ValueError:
        print("Error! Please insert only numbers...")
        x = input("Insert numbers only: ")

print(x/2)
```

#### Output 1

```
Insert numbers only: er56
Error! Please insert only numbers...
Insert numbers only:
```

#### Output 2

```
Insert numbers only: 4589lk
Error! Please insert only numbers...
Insert numbers only:
```

#### Output 3

```
Insert numbers only: 5678
2839.0
```

**Practice 85: A Program to Check for Non-Existence of Number**

```
x = 20
```

```
y = 30
```

```
z = 40
```

```
value = input("Insert variable x,y,z only: ")
```

```
try:
```

```
    exec("print("+value+")")
```

```
except NameError:
```

```
    print("Incorrect variable name!")
```

**Output 1**

```
Insert variable x,y,z only: y
```

```
30
```

**Output 2**

```
Insert variable x,y,z only: z
```

```
40
```

**Output 3**

```
Insert variable x,y,z only: z
```

```
40
```

**Output 4**

```
Insert variable x,y,z only: d
```

```
Incorrect variable name!
```

**Practice 86: A Program to Display Error Message for Non-Existed Files**

```
fp = input("Enter file path: ")

try:
    file = open(fp)
except FileNotFoundError:
    print("Error! This file path does not exist...")
else:
    print(file.read())
```

**Output 1**

```
Enter file path: fhh
Error! This file path does not exist...
```

**Output 2**

```
Enter file path: C:\Users\USER\Desktop\GENERAL\File\mytest.txt
Welcome to computer programming
```

### Practice 87: A Program to Get Division by Zero Error

```
x = float(input("Insert first number: "))
y = float(input("Insert second number: "))

try:
    z = x/y
except ZeroDivisionError:
    print("Error! Number not divisible by zero...")
else:
    print(z)
```

#### Output 1

```
Insert first number: 7
Insert second number: 5
1.4
```

#### Output 2

```
Insert first number: 8
Insert second number: 0
Error! Number not divisible by zero...
```

#### Output 3

```
Insert first number: 19
Insert second number: 0
Error! Number not divisible by zero...
```

### Practice 88: A Program to Get Index Out of Exception

```
x = [1,2,3,4,5,6,7,8,9]

print("t = terminate")

while True:
    num = input("Choose index to search: ")
    if num == 't':
        break
    try:
        num = int(num)
        print(x[num])
    except ValueError:
        print("Only integers are allowed!")
    except IndexError:
        print("Error! Number out of index", num)
```

#### Output 1

```
t = terminate
Choose index to search: 17
Error! Number out of index 17
Choose index to search: t
```

#### Output 2

```
t = terminate
Choose index to search: 6
7
Choose index to search: t
```

#### Output 3

```
t = terminate
Choose index to search: z
Only integers are allowed!
Choose index to search:
```



### Practice 89: A Program to Raise Exceptions

```
x = ['a','b','c','d','e']
y = input("Insert a letter: ")

if y in x:
    print(1)
else:
    raise ValueError("Letter does not exist!")
```

#### Output

```
Insert a letter: f
Traceback (most recent call last):
  File "C:/Users/USER/Python Projects/Project 89.py", line 7, in <module>
    raise ValueError("Letter does not exist!")
ValueError: Letter does not exist!
```

TECHVIDYA

### Practice 90: A Program to Use Classes and Constructor

```
class company:
    def __init__(self, employee_firstname, employee_surname):
        self.firstname = employee_firstname
        self.surname = employee_surname

    def employee(self):
        return self.firstname + ' ' + self.surname

staff = []
for i in range(1):
    name = input("Insert firstname and surname: ")
    name = name.split()
    staff.append(company(name[0], name[1]))

for i in staff:
    print(i.employee())
```

#### Output 1

```
Insert firstname and surname: Peter Paul
Peter Paul
```

#### Output 2

```
Insert firstname and surname: Hello World
Hello World
```

#### Output 3

```
Insert firstname and surname: Helen Diana
Helen Diana
```

**Practice 91: A Program to Fill a List with Natural Numbers**

```
x = int(input("Insert range of natural numbers: "))  
num = [y+1 for y in range(x)]  
print(num)
```

**Output 1**

```
Insert range of natural numbers: 10  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10]
```

**Output 2**

```
Insert range of natural numbers: 15  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15]
```

**Output 3**

```
Insert range of natural numbers: 20  
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20]
```

### Practice 92: A Program to Fill a List with Random Numbers

```
from random import randint

mini = int(input("Insert minimum number: "))
maxi = int(input("Insert maximum number: "))

y = int(input("Insert range of random numbers: "))

x = [randint(mini,maxi) for i in range(y)]

print(x)
```

#### Output 1

```
Insert minimum number: 4
Insert maximum number: 10
Insert range of random numbers: 15
[8, 8, 6, 6, 9, 8, 10, 9, 8, 8, 10, 7, 9, 9, 5]
```

#### Output 2

```
Insert minimum number: 1
Insert maximum number: 15
Insert range of random numbers: 20
[13, 1, 1, 1, 12, 13, 13, 13, 1, 10, 4, 9, 11, 5, 8, 8, 1, 6, 11, 14]
```

#### Output 3

```
Insert minimum number: 2
Insert maximum number: 6
Insert range of random numbers: 10
[5, 2, 6, 5, 3, 5, 2, 3, 2, 2]
```

**Practice 93: A Program to Group Dictionaries into Lists**

```
dict = {'one':1, 'two':2, 'three':3}
```

```
x = [[y,z]for y, z in dict.items()]
```

```
print(x)
```

**Output**

```
[['one', 1], ['two', 2], ['three', 3]]
```





**Practice 94: A Program to Unpack Matrix into One Level List**

```
x = [[1,2,3],[4,5,6],[7,8,9]]
```

```
y = [item for row in x for item in row]
```

```
print(y)
```

**Output 1**

```
[[1, 2, 3], [4, 5, 6], [7, 8, 9]]
```

```
[1, 2, 3, 4, 5, 6, 7, 8, 9]
```



**Practice 95: A Program to Read Data from File into Lines and Characters**

```
x = [i.strip() for i in  
open("C:/Users/USER/Desktop/PythonProjects/Python_File/Python.txt")]  
  
print(x)
```

**Output**

**Change the file path below to your file location**

**C:/Users/USER/Desktop/Python Projects/Python\_File/Python.txt**



### Practice 96: A Program to Calculate the Orbital Speed

```
from math import pi

r = float(input("Insert radius of the orbit(million km): "))
v = float(input("Insert orbital speed(km/s): "))

r = r*1000000

yr = 2*pi*r/v

yr = yr/(60*60*24)

print(round(yr))
```

#### Output 1

```
Insert radius of the orbit(million km): 10
Insert orbital speed(km/s): 35
21
```

#### Output 2

```
Insert radius of the orbit(million km): 50
Insert orbital speed(km/s): 83
44
```

#### Output 3

```
Insert radius of the orbit(million km): 100
Insert orbital speed(km/s): 45
162
```

**Practice 97: A Program to Get Sum of Random of 3 Digit Numbers**

```
from random import random

x = random()*900+100
x = int(x)
print('Three random numbers =',x)

s = str(x)

y = int(s[0])*100
z = int(s[1])*10
w = int(s[2])

print('Sum of the random numbers =',y+z+w)
```

**Output 1**

```
Three random numbers = 619
Sum of the random numbers = 16
```

**Output 2**

```
Three random numbers = 601
Sum of the random numbers = 7
```

**Output 3**

```
Three random numbers = 121
Sum of the random numbers = 4
```

**Practice 98: A Program to Select Even Numbers from List**

```
x = [1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16]
```

```
y = [i for i in x if i % 2 == 0]
```

```
print(y)
```

**Output**

```
[1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16]  
[2, 4, 6, 8, 10, 12, 14, 16]
```





### Practice 99: A Program to Calculate the Area and Perimeter of Right-Angle Triangle

```
import math

x = float(input("Insert length of x: "))
y = float(input("Insert length of y: "))

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

Area = (x*y)/2
Perimeter = x+y+z
print("Area of right angled triangle = %.2f" % Area)
print("Perimeter of right angled triangle = %.2f" % Perimeter)
```

#### Output 1

```
Insert length of x: 15
Insert length of y: 7
Area of right angled triangle = 52.50
Perimeter of right angled triangle = 38.55
```

#### Output 2

```
Insert length of x: 10
Insert length of y: 19
Area of right angled triangle = 95.00
Perimeter of right angled triangle = 50.47
```

#### Output 3

```
Insert length of x: 12
Insert length of y: 9
Area of right angled triangle = 54.00
Perimeter of right angled triangle = 36.00
```

### Practice 100: A Program to Get Total Surface Area of a Cylinder

```
from math import pi

h = float(input("Insert height of cylinder: "))
r = float(input("Insert radius of cylinder: "))

circles = 2*(pi*r**2)
side = 2*pi*r*h
Area = circles+side

print("Total surface area = ",round(Area,2))
```

#### Output 1

```
Insert height of cylinder: 10
Insert radius of cylinder: 5
Total surface area = 471.24
```

#### Output 2

```
Insert height of cylinder: 20
Insert radius of cylinder: 17
Total surface area = 3952.12
```

#### Output 3

```
Insert height of cylinder: 15
Insert radius of cylinder: 8
Total surface area = 1156.11
```

## Practice 101: A Complete Program to Build A Working Digital Electronic Calculator

```

from tkinter import *

def btn(numbers):
    global operator
    operator = operator + str(numbers)
    txt_input.set(operator)

def Clear():
    global operator
    operator=""
    txt_input.set("")
    Display.insert(0,'Start Calculating...')

#def Equal():
#    global operator
#    sumu = float(eval(operator))
#    txt_input.set(sumu)
#    operator=""

def Equal():
    global operator
    try:
        global operator
        sumu = float(eval(operator))
    except ZeroDivisionError:
        txt_input.set("Zero Error")
    else:
        txt_input.set(sumu)
        operator = ""

root = Tk()
root.title('Calculator')

```

```

operator=""
txt_input = StringVar(value='Start Calculating...')

#=====Screen=====
=====
Display = Entry(root,font=('arial',30,'bold'), fg='white', bg='green',
                justify='right',bd=50,textvariable=txt_input)
Display.grid(columnspan=4)

#=====Row1=====
=====
btn7 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='7',command=lambda:btn(7)).grid(row=1,column=0)
btn8 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='8',command=lambda:btn(8)).grid(row=1,column=1)
btn9 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='9',command=lambda:btn(9)).grid(row=1,column=2)
btnC = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='C',bg='green',command=Clear).grid(row=1,column=3)

#=====Row2=====
=====
btn4 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='4',command=lambda:btn(4)).grid(row=2,column=0)
btn5 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='5',command=lambda:btn(5)).grid(row=2,column=1)
btn6 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='6',command=lambda:btn(6)).grid(row=2,column=2)
btnplus = Button(root,padx=33,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
                 text='+',bg='orange',command=lambda:btn('+')).grid(row=2,column=3)

#=====Row3=====
=====
btn1 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
              text='1',command=lambda:btn(1)).grid(row=3,column=0)
btn2 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),

```



```

        text='2',command=lambda:btn(2)).grid(row=3,column=1)
    btn3 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
        text='3',command=lambda:btn(3)).grid(row=3,column=2)
    btnminus = Button(root,padx=38,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
        text='-',bg='orange',command=lambda:btn('-')).grid(row=3,column=3)

#=====Row4=====
btn0 = Button(root,padx=30,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text='0',command=lambda:btn(0)).grid(row=4,column=0)
btndot = Button(root,padx=35,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text='.',bg='orange',command=lambda:btn('.')).grid(row=4,column=1)
btndivision = Button(root,padx=36,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text='/',bg='orange',command=lambda:btn('/')).grid(row=4,column=2)
btnmultiply =
Button(root,padx=34,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text='x',bg='orange',command=lambda:btn('*')).grid(row=4,column=3)

#=====Row5=====
btnequals = Button(root,padx=95,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text='=',bg='green',command=Equal).grid(row=5,column=0,columnspan=2)
btnopenbracket =
Button(root,padx=35,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text='(',bg='orange',command=lambda:btn('(')).grid(row=5,column=2)
btnclosebracket =
Button(root,padx=38,pady=15,bd=8,fg='black',font=('arial',30,'bold'),
    text=')',bg='orange',command=lambda:btn(')')).grid(row=5,column=3)

root.mainloop()

```







# TECHVIDYA

ISO 9001:2015 Accredited Company

*"Stay Updated, Stay Ahead"*

For TechVidya Candidates Only.  
Not For Selling Purpose.