

Python is a widely used high-level programming language. To write and execute code in python, we first need to install Python on our system.

Installing Python on Windows takes a series of few easy steps.

Step 1: Download the Full Installer

Follow these steps to download the full installer:

- Open a browser window and navigate to the Python.org Downloads page for Windows.
- Under the “Python Releases for Windows” heading, click the link for the Latest Python 3 Release - Python 3.x.x. As of this writing, the latest version was Python 3.10.
- Scroll to the bottom and select either Windows x86-64 executable installer for 64-bit or Windows x86 executable installer for 32-bit.

Step 2 – Run Executable Installer

We downloaded the Python 3.10 Windows 64 bit installer.

Run the installer. Make sure to select both the checkboxes at the bottom and then click Install New.



On clicking the Install Now, The installation process starts.



The installation process will take few minutes to complete and once the installation is successful, the following screen is displayed.

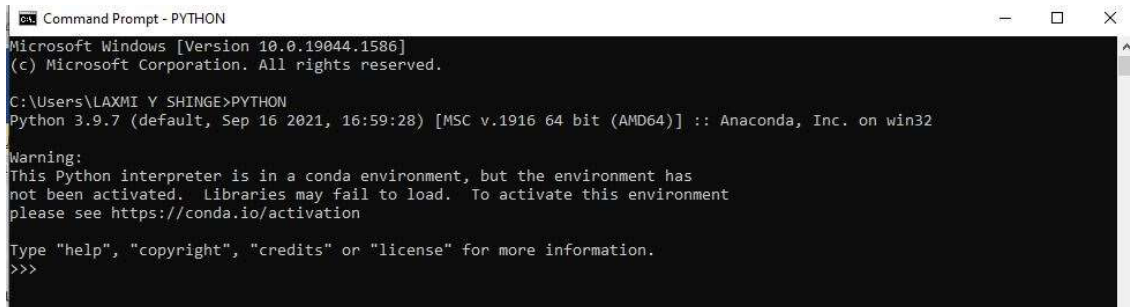


Step 3 – Verify Python is installed on Windows

To ensure if Python is successfully installed on your system. Follow the given steps

- Open the command prompt.
- Type 'python' and press enter.

The version of the python which you have installed will be displayed if the python is successfully installed on your windows.



```
Command Prompt - PYTHON
Microsoft Windows [Version 10.0.19044.1586]
(c) Microsoft Corporation. All rights reserved.

C:\Users\LAXMI Y SHINGE>PYTHON
Python 3.9.7 (default, Sep 16 2021, 16:59:28) [MSC v.1916 64 bit (AMD64)] :: Anaconda, Inc. on win32

Warning:
This Python interpreter is in a conda environment, but the environment has
not been activated. Libraries may fail to load. To activate this environment
please see https://conda.io/activation

Type "help", "copyright", "credits" or "license" for more information.
>>>
```

PyCharm is a cross-platform editor developed by JetBrains. PyCharm provides all the tools you need for productive Python development.

Below are the detailed steps for installing PyCharm

Step1: To download PyCharm visit the website <https://www.jetbrains.com/pycharm/download/> and Click the “DOWNLOAD” link under the Community Section.

Download PyCharm



Windows macOS Linux

Professional

Full-featured IDE for Python & Web development

DOWNLOAD

Free trial

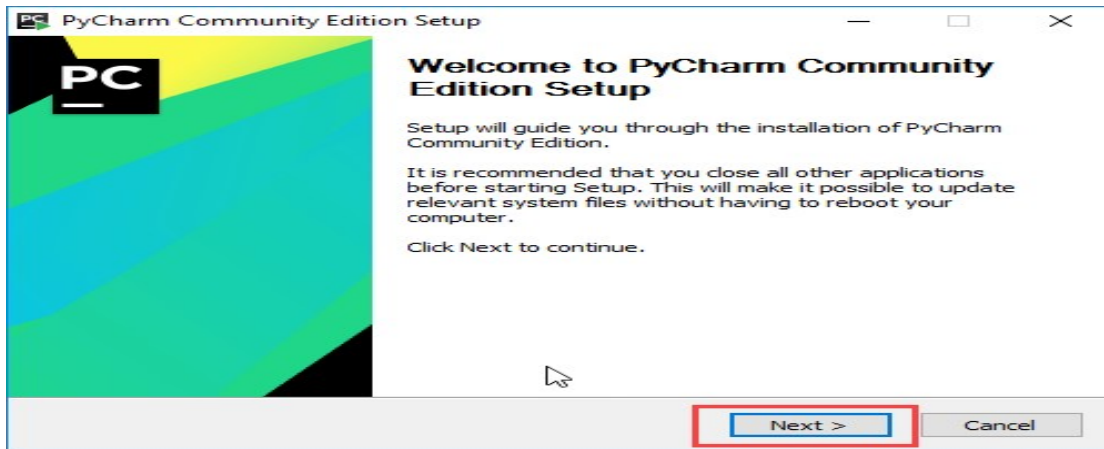
Community

Lightweight IDE for Python & Scientific development

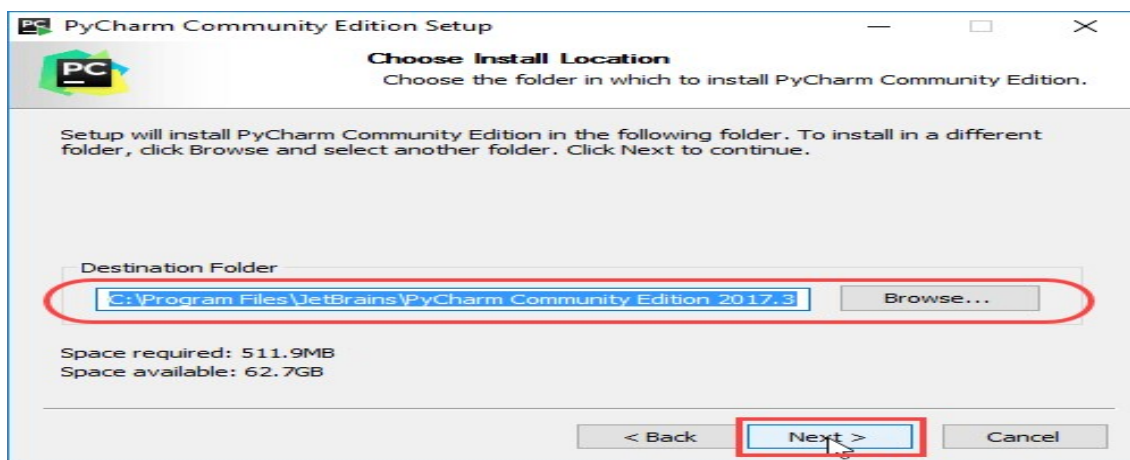
DOWNLOAD

Free, open-source

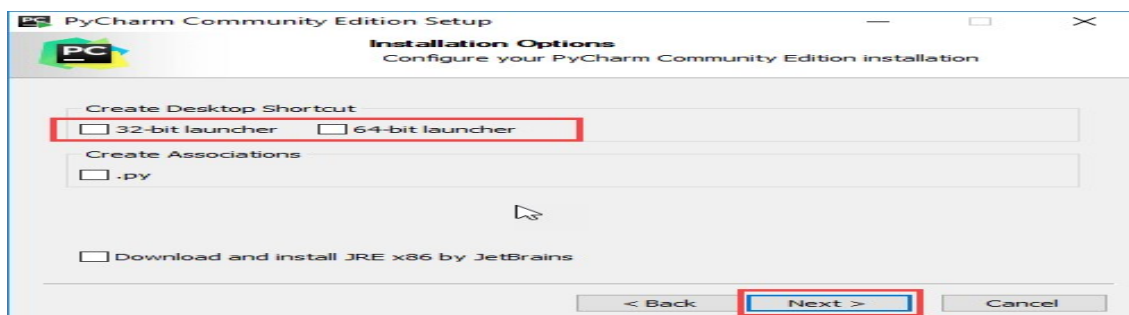
Step 2) Once the download is complete, run the exe for install PyCharm. The setup wizard should have started. Click “Next”.



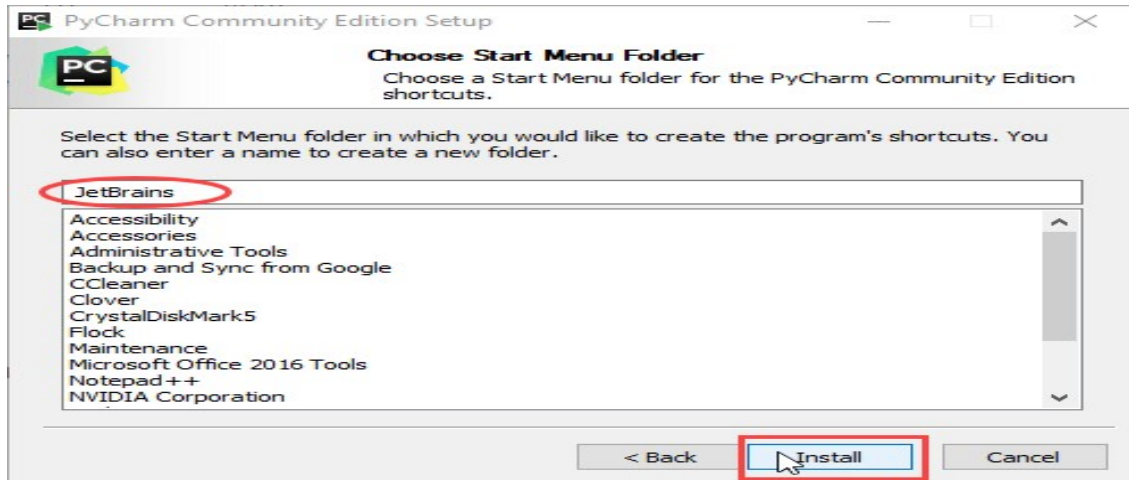
Step 3) On the next screen, Change the installation path if required. Click “Next”.



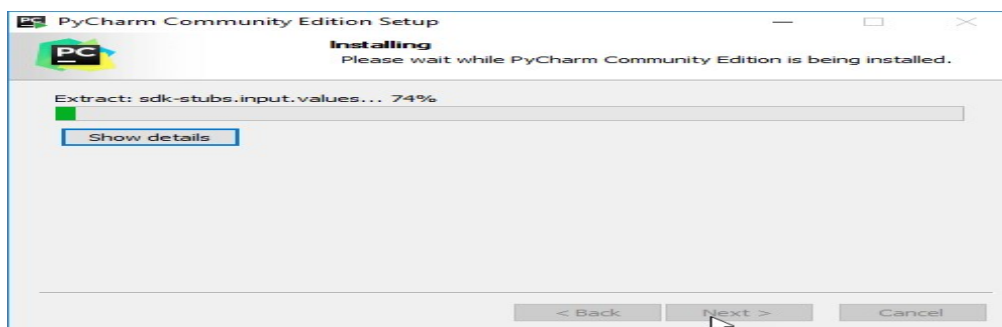
Step 4) On the next screen, you can create a desktop shortcut if you want and click on “Next”.



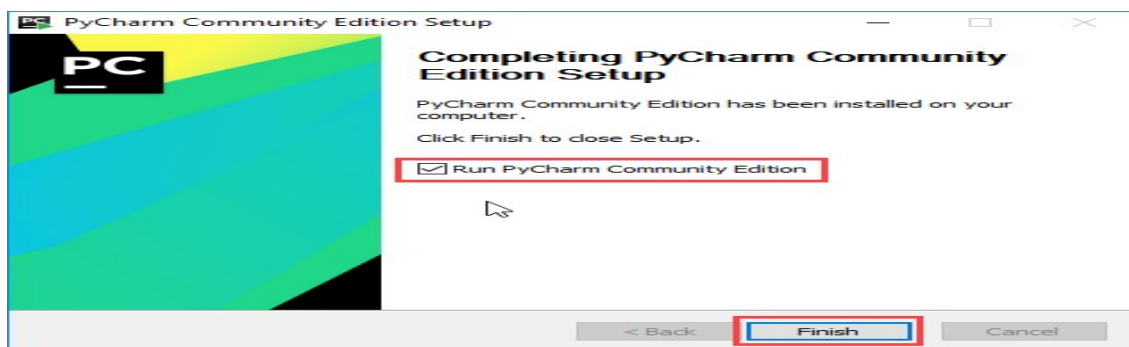
Step 5) Choose the start menu folder. Keep selected JetBrains and click on “Install”.



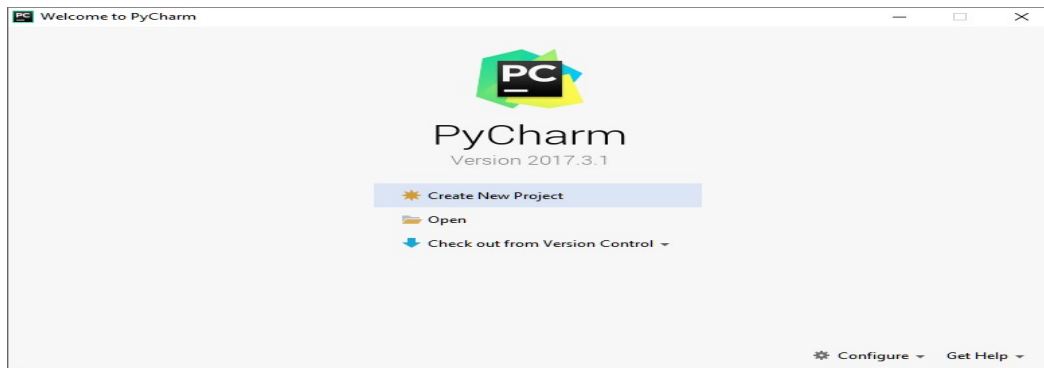
Step 6) Wait for the installation to finish.



Step 7) Once installation finished, you should receive a message screen that PyCharm is installed. If you want to go ahead and run it, click the “Run PyCharm Community Edition” box first and click “Finish”.



Step 8) After you click on “Finish,” the Following screen will appear.



- 1. Write a Program to illustrate the use of I/O statements**
Input the details of students such as Register Number, Name, semester ,Percentage and college name from the keyboard and display the same information on the screen
[hint: use input() and print()]

Algorithm:

Step1: Read Register Number, Student name, semester, Percentage and college name from the keyboard Using input()

Step2: Display the same information on the screen using print()

Program:

```
Register_Number=input("Enter your register number:")
Student_Name=input("Enter your name:")
Sem=int(input("Enter your semester:"))
Percentage=float(input("Enter your percentage:"))
College_Name=input("Enter your college name:")
print("\n\n\n\n")

print("Register Number:",Register_Number)
print("Student Name :",Student_Name)
print("Semester:",sem)
```

```
print("Percentage:",Percentage)
```

```
print("Name of the college is:",College_Name)
```

2. Write a program to Evaluate expressions and displays formatted output.

Algorithm:

Step 1: Read two numbers from the user as a and b

Step 2: Read an expression from keyboard

Step 3: Calculate sum and difference of these values and store these result in variables called sum and sub respectively.

Step 3: Display the values of a, b, sum, sub using string format function.

Step 4: Display the result of expression

(a) string formatting using format() function

```
a = int(input("Enter a:"))
b = int(input("Enter b:"))
expression=4+5-6*(4+5)/2(6-1)
sum=a+b
sub=a-b
print('The value of a is { } and b is { }'.format(a,b))
print('{2} is the sum of {0} and {1}'.format(a,b,sum))
print('{2} is the subtraction of {0} and {1}'.format(a,b,sub))
print("The result of expression is{ } :".format(expression))
```

3. Code, execute and debug the programs to illustrate the decision making statements.

a) Read a number and check whether its positive(simple if)

Algorithm:

Step1: Read the number

Step2: Check whether the number is greater than 0 then

display “ The number is positive”

Program:

```
Number=int(input("Enter the number:"))
```

```
if Number>0:
```

```
    print(f'{Number} is positive')
```

b) Read the age of person and check whether he is eligible to vote or not (use of if.. else statement)

Algorithm:

Step1: Read the age

Step 2: Check if the age is greater than or equal to 18 then

Display “ Eligible to cast the vote”

Step 3: Otherwise display “ Not eligible to cast the vote”

Program:

```
Age=int(input("Enter the age of a person:"))
```

```
if Age>=18:
```

```
    print("You are eligible to cast the vote")
```

```
else:
```

```
    print("You are not eligible")
```


c) If the ages of Ram, Shyam and Ajay are input through the keyboard, write a program to determine the elder of the three.(if..elif..else)

Algorithm:

Step1: Take a input for ages of Ram,Shyam and Ajay.

Step2: If Ram's age is greater than Shyam age and Ajay age then
display "Ram is elder".

Step3: Otherwise,If Shyam age is greater than Ram age and Ajay age then
display "Shyam is Elder"

Step4: Otherwise display "Ajay is elder".

Program:

```
ram=int(input("Enter a age of Ram:"))
shyam=int(input("Enter a age of Shyam:"))
ajay=int(input("Enter a age of Shyam:"))
if (ram>shyam)and ( ram>ajay):
    print("Ram is Elder.")
elif shyam>ram and shyam>ajay:
    print("Shyam is Elder.")
else:
    print("Ajay is Elder.")
```

d) Take a year as an input from the user and determine whether its leap year or not

Algorithm:

Step1: Read year

Step2: Check if the year is perfectly divisible by 100 and 400 then
display "Year is a leap year"

Step 3: If the year is perfectly divisible by 100 and not by 400 then

display “ year is not a leap year”

Step 4: If the year is not divisible by 100 then check if the year is divisible by 4

Then the year is a leap year

Step 5: Otherwise the year is not a leap year

Program:

```
year=int(input("Enter the year:"))
if year%100==0:
    if year%400==0:
        print(f“ {year} is a leap year”)
    else:
        print(f“ {year} is not a leap year”)
else:
    if year%4==0:
        print(f“ {year} is a leap year”)
    else:
        print(f“ {year} is not a leap year”)
```

4 . Code and execute python programs to illustrate the use of loops

- a) Program to check whether the given number is Palindrome or not

Algorithm:

Step1 : Read the number

Step 2: Initialize Reverse to 0

Step 3: Assign number to Original number as Original_number=number

Step 4 : As long as the number is not zero repeat the following steps

Extract the digit from the number i.e, $rem = number \% 10$

Find the reverse number i.e, $rev_number = (rev_number * 10) + rem$

Get the number after digit extraction i.e, $number = number // 10$

Step 5: If the Original_number is equal to the rev_number then

Display “The number is palindrome”

Step 6: Otherwise Display “The number is not palindrome”

Program:

```
Number=int(input("Enter the number:"))
```

```
Reverse_Number=0
```

```
Original_Number=Number
```

```
while Number!=0:
```

```
    rem=Number%10
```

```
    Reverse_Number=(Reverse_Number*10)+rem
```

```
    Number=Number//10
```

```
if Original_Number==Reverse_Number:
```

```
    print(f'{Original_Number} is palindrome')
```

```
else:
```

```
    print(f'{Original_Number} is not palindrome')
```

- b) Program to check whether a given number is prime or not.

Algorithm:

Step1: Read the number

Step2: Initialize Is_prime to True

Step3: A prime number is a number which is divisible by 1 and itself. Use a for loop to determine whether a number is divisible by any number from 2 to number. If so then set Is_prime to False and break out of loop.

Step 4: Now check if Is_prime is True then

Display “The number is prime”

Step 5: Otherwise Display “The number is not prime”

Program:

```
Number=int(input("Enter the number:"))
Is_prime=True
for i in range(2,Number):
    if Number%i==0:
        Is_prime=False
        Break
if Is_prime:
    print(f'{Number} is a prime number')
else:
    print(f'{Number} is not a prime number')
```

- c) Write a program to find the factorial value of any number entered through the keyboard.

Note: Use While Loop.

Algorithm:

Step1: Input a number from the keyboard.
Step2: Initialize the result to 1 and store it to one variable.
Step3: Start a loop and multiply the result by the number.
Step4: Reduce one from the number in each iteration.
Step5: End the loop once the number reaches 1
Step6: display the result value.

Program:

```
n=int(input("Enter a number:"))
temp=n
fact=1

while n!=0:
    fact=fact*n
    n=n-1
print("Factorial of {0} is {1}.".format(temp,fact))
```

5. Code and execute python programs to illustrate set and tuple data types

a) Write a program to create two sets using set comprehension and perform following set operations:

- (i) Union
- (ii) Difference
- (iii) Symmetric Difference
- (iv) Intersection.

Algorithm:

Step1: Create a two sets.

Step2: Perform a union operation using | operator and display it's result.

Step3: Perform a difference operation using - operator and display it's result.

Step4: Perform a symmetric difference operation using ^ operator and display it's result.

Step5: Perform a intersection operation using & operator and display it's result.

Program:

```
set1={var**2 for var in range(1,11)}
set2={i for i in range(1,11) if i%2==0}
print("Elements of Set1 are :",set1 )
print("Elements of Set2 are :",set2 )

print("Union Operation",set1|set2)
print("Intersection Operation",set1&set2)
print("Difference Operation",set1-set2)
print("Symmetric Difference Operation",set1^set2)
```

b) Write a program to create a tuple and perform following operations

- (i) Display the elements of tuple
- (ii) Find an item using index method
- (iii) Reverse all the elements
- (iv) Display the elements from 3 rd position to 7th position
- (v) Delete entire tuple

ALGORITHM:

Step 1: Start
Step 2: Declare and Initialize tuple1
Step 3: Display tuple1
Step 4: Find an item using index method.
Step 5: Reverse tuple1 elements and Display
Step 6: Display elements from 3rd to 7th position
Step 7: Delete tuple1
Step 8: Stop

PROGRAM:

```
tuple1=(1,2,3,4,5,6,7,8,9)
print("Elements of tuple are:\n")
for i in range(len(tuple1)):
    print(tuple1[i])
find=int(input("Enter element to search in tuple "))
i=tuple1.index(find)
print("\n Element %d found at index %d in tuple \n"%(find,i))
print("Elements of tuple in reverse order \n ",tuple1[::-1])
print("Elements from 3 rd position to 8 th position",tuple1[3:8])
```

6 . Write a program to create a List of 10 odd numbers using List Comprehension and perform the following:

- i) Display all the elements
- (ii) Find the length of list
- (iii) Adding new items to list using append(), insert()
- (iv) Remove certain items using pop(), remove()
- (v) Find the particular item using index()

ALGORITHM:

Step 1: Start

Step 2: Declare and initialize List1 using Comprehension

Step 3: Display List1

Step 4: Obtain the length of List1

Step 5: Add items using append(), insert() to List1

Step 6: Remove items using pop(), remove() from List1

Step 7: Find the particular item using index() and display

Step 8: Stop

PROGRAM:

```
list1=[i for i in range(20) if i%2!=0]
print("Elements of List are ")
for i in range(len(list1)):
    print(list1[i])
print("Length of list is ", len(list1))
print("Adding new item using append method")
list1.append(99)
print("Adding new item using insert method")
list1.insert(1,2)
print("Elements are: ", list1)
print("Removing an item using pop method")
list1.pop(1)
print("Removing an item using remove method")
list1.remove(3)
print("Elements are: ", list1)
find=int(input("Enter the element to search"))
pos=list1.index(find)
print("%d element found at index %d"%(find,pos))
```

7 Create a dictionary that contains States and their capital city as the associated values. Make up the data for five dictionary entries and demonstrate the use of built-in function and methods.

Algorithm:

Step1: Start

Step2: Create empty dictionary

Step3: Read the total of number of entries to be made in the dictionary

Step4: Construct the dictionary incrementally by reading the key and value from the user using for loop and update()

Step5: Apply all built-in functions on the dictionary

Step6: Apply all built-in methods on the dictionary

Step 7: End

Program:

```
Construct_dict={}
Total_No_of_items=int(input("Enter number of dictionary entries:- "))
for i in range(Total_No_of_items):
    dict_key=input("Enter the key:-")
    dict_value=input("Enter the value:-")
    Construct_dict.update({dict_key:dict_value})
print("The constructed dictionary is:", Construct_dict)
print("The no of data items in the dictionary are:-",len(Construct_dict))
print("The keys of dictionary in asc order:-",sorted(Construct_dict))
print("Key in descending order:-", sorted(Construct_dict,reverse=True))
print("Arranging the values of dictionary in ascending order:-")
print(sorted(Construct_dict.values()))
print("Arranging the data items in ascending order:-")
print(sorted(Construct_dict.items()))
```



```
states=Construct_dict.fromkeys(Construct_dict)
print("The contents of states dictionary is",states)
# fromkeys() creates a new dictionary from sequence with default value
states=Construct_dict.fromkeys(Construct_dict,"BENGALURU")
print("The contents of states dictionary is",states)

print("Use of get():",Construct_dict.get("Karnataka"))
print("Use of get():",Construct_dict.get("TamilNadu"))
print("Printing the keys from dictionaries:-\n",Construct_dict.keys())
print("Printing the values from dictionaries:-\n",Construct_dict.values())
print("Printing the data-itemsfrom dictionaries:-\n",Construct_dict.items())
print("Updating the dictionary with update() method:-")
Construct_dict.update({"TamilNadu":"Chennai"})
print("The updated contents of dictionary:--",Construct_dict)
Construct_dict.setdefault("Uttarpradesh")
print("Use of setdefault() method",Construct_dict)
Construct_dict.setdefault("Bihar","Patna")
print("Use of setdefault() method with specified value",Construct_dict)
print("The pop() method returns the specified key and returns its value:")
print(Construct_dict.pop("Uttarpradesh"))
print("pop the data item:-")
print(Construct_dict.popitem())
```

Output:

Enter number of dictionary entries:- 2

Enter the key:-Karnataka

Enter the value:-Bengaluru

Enter the key:-Maharashtra

Enter the value:-Mumbai

The constructed dictionary is: {'Karnataka': 'Bengaluru', 'Maharashtra': 'Mumbai'}

The no of data items in the dictionary are:- 2

The keys of dictionary in asc order:- ['Karnataka', 'Maharashtra']

Key in descending order:- ['Maharashtra', 'Karnataka']

Arranging the values of dictionary in ascending order:-

['Bengaluru', 'Mumbai']

Arranging the data items in ascending order:-

[('Karnataka', 'Bengaluru'), ('Maharashtra', 'Mumbai')]

The contents of states dictionary is {'Karnataka': None, 'Maharashtra': None}

The contents of states dictionary is {'Karnataka': 'BENGALURU', 'Maharashtra': 'BENGALURU'}

Use of get(): Bengaluru

Use of get(): None

Printing the keys from dictionaries:-

dict_keys(['Karnataka', 'Maharashtra'])

Printing the values from dictionaries:-

dict_values(['Bengaluru', 'Mumbai'])

Printing the data-items from dictionaries:-

dict_items([('Karnataka', 'Bengaluru'), ('Maharashtra', 'Mumbai')])

Updating the dictionary with update() method:-

The updated contents of dictionary:-- {'Karnataka': 'Bengaluru', 'Maharashtra': 'Mumbai', 'TamilNadu': 'Chennai'}

Use of setdefault() method {'Karnataka': 'Bengaluru', 'Maharashtra': 'Mumbai', 'TamilNadu': 'Chennai', 'Uttarpradesh': None}

Use of setdefault() method with specified value {'Karnataka': 'Bengaluru', 'Maharashtra': 'Mumbai', 'TamilNadu': 'Chennai', 'Uttarpradesh': None,

```
'Bihar': 'Patna'}
```

The pop() method returns the specified key and returns its value:

None

pop the data item:-

```
('Bihar', 'Patna')
```

8a. Consider Two strings S1 “Government Polytechnic” and S2 “Athani,Belagavi” Create a program to create these 2 strings and perform the following:

- (i) Find First character in S1
- (ii) Find Last but one character in S2
- (iii) Find length of both strings
- (iv) Reverse S1
- (v) Apply center(), zfill(), lower(), upper(), replace(), find(), join(), max(), min() methods

ALGORITHM:

Step 1: Start

Step 2: Declare S1 and assign “Government polytechnic” to S1

Step 3: Declare S2 and assign “Athani, Belagavi” to S2

Step 4: Display first character of S1

Step 5: Display last character of S2

Step 6: Obtain length of S1 and display

Step 7: Obtain length of S2 and display

Step 8: Declare variable reverse

Step 9: Use a for loop to iterate through S1

Step 10: Reverse S1 string and assign to reverse variable

Step 11: Display reverse

Step 12: Declare a string list, assign to str

Step 13: Use join() to join str , assign to str2

Step 14: Display str2

Step 15: Apply center() and display

Step 16: Apply zfill() and display

Step 17: Apply lower() and display

Program:

```
S1 = "Government Polytechnic"
S2 = "Athani,Belagavi"
print("String1 is ", S1)
print("String2 is ", S2)
print("First character of S1: ",S1[0])
print("Last but one character of S2: ",S2[-2])
print("Length of String1 is:", len(S1))
print("Length of String2 is:",len(S2))

reverse = ""
for i in S1:
    reverse = i + reverse
print("Reversed string of String1",reverse)
print("Demonstration of String Operations")
str = ["Strings", "In", "Python"]
str2=(" ").join(str)
print(str2)
print("Center() method:",S1.center(30,"*"))
print("zfill method:",S1.zfill(35))
print("Lowercase version of s1:",S1.lower())
print("Uppercase version of s1:",S1.upper())
print("Find Method:",S1.find("P"))
print("Max Method:",max(S1))
print("Min Method:",min(S2))
S = S1.replace("Government Polytechnic" , "Diploma college")
print(S)
```

Output:

String1 is Government Polytechnic

String2 is Athani,Belagavi

First character of S1: G

Last but one character of S2: v

Length of String1 is: 22

Length of String2 is: 15

Reversed string of String1 cinhcetyloP tnemnrevoG

Demonstration of String Operations

Strings In Python

Center() method :****Government Polytechnic****

zfill method: 0000000000000Government Polytechnic

Lowercase version of s1 government polytechnic

Uppercase version of s1 GOVERNMENT POLYTECHNIC

Find Method: 11

Max Method: y

Min Method: ,

Diploma college

8b. Write a program to create two arrays (i) integer numbers array (ii) floating point numbers array and perform following operations:

- (a) Insert new elements using insert()**
- (b) Remove Existing elements using pop(), remove()**
- (c) print elements from beginning to a range use [:Index].**
- (d) print elements from end use [:-Index]**
- (e) print elements from specific Index till the end use [Index:]**
- (f) print elements within a range, use [Start Index : End Index]**
- (g) print whole List with the use of slicing operation, use [:].
print whole array in reverse order, use [::-1].**

ALGORITHM:

Step 1: Start

Step 2: Import array module

Step 3: Declare array of integer numbers and assign to ary1

Step 4: Declare array of floating point numbers and assign to ary2

Step 5: Insert an item using insert() method

Step 6: Remove Existing elements using pop() , remove()

Step 7: Display elements from beginning to a range

Step 8: Display elements from end

Step 9: Display elements from specific Index to till the end

Step 10: Display elements within a range

Step 11: Display ary2

Step 12: Display ary2 in reverse order

Step 13: Stop

PROGRAM:

```
import array as arr
ary1 = arr.array('i' , [1,2,3,4,5,6,7,8,9])
ary2 = arr.array('d' , [1.1,2.5,6.7,3.5,2.8,9.1])
print("Elements of First array are :")
for i in range(len(ary1)):
    print(ary1[i])
print("Elements of Second array are :")
for i in range(len(ary2)):
    print(ary2[i])
print("Insertion")
ary1.insert(0 , 10)
print("Updated array is ",ary1)
print("Removal")
ary1.remove(10)
print("Updated array is ",ary1)
print(" pop() removes element from the specified index:", ary1.pop(8))
print("Updated array is ",ary1)
print("Slicing Operation")
print(ary1[ : len(ary1)])
print(ary1[-1 : ])
print(ary1[2 : ])
print(ary1[2 : 8])
print(ary2[ : ])
print(ary2[ : -1])
```


Output:

Elements of First array are :

1
2
3
4
5
6
7
8
9

Elements of Second array are :

1.1
2.5
6.7
3.5
2.8
9.1

Insertion

Updated array is `array('i', [10, 1, 2, 3, 4, 5, 6, 7, 8, 9])`

Removal

Updated array is `array('i', [1, 2, 3, 4, 5, 6, 7, 8, 9])`

`pop()` removes element from the specified index: 9

Updated array is `array('i', [1, 2, 3, 4, 5, 6, 7, 8])`

Slicing Operation

`array('i', [1, 2, 3, 4, 5, 6, 7, 8])`

`array('i', [8])`

`array('i', [3, 4, 5, 6, 7, 8])`

`array('i', [3, 4, 5, 6, 7, 8])`

`array('d', [1.1, 2.5, 6.7, 3.5, 2.8, 9.1])`

`array('d', [9.1, 2.8, 3.5, 6.7, 2.5, 1.1])`

9 a) Read a number from the user. Create CheckArmstrong() to determine whether given number is Armstrong or not.

Algorithm:

Step 1: Define a function CheckArmstrong() using def keyword.

Step 2: Initialize sum to 0

 Assign the number to temp

 As long as the number is not equal to 0 then

 Extract the digit

 Find the cube of the digit and add it with sum

 Retain the remaining number after extraction

Step 3: Read a number

Step 4: Make a call to CheckArmstrong() to determine the output

Step 5: Stop

Program:

```
def CheckArmstrong(number):  
    sum=0  
    temp=number  
    while temp!=0:  
        rem=temp%10  
        sum=sum+rem**3  
        temp=temp//10  
    if number==sum:  
        print(f"{number} is Armstrong")  
    else:  
        print(f"{number} is not a Armstrong")
```

Driver code to call the function

CheckArmstrong(153)

CheckArmstrong(916)

Output

153 is a Armstrong

916 is not a Armstrong

9b) For any Given positive integer find its factorial using recursion**Algorithm:**

Step 1: Define a function Factorial() using def keyword.

Step 2: if the number is equal to 0 then return 1

Otherwise return number*Factorial(number-1)

Step 3: Read a number

Step 3: Make a call to Factorial() to determine the output

Step 4: Stop

Program:

Function Defintion:

```
def Factorial(n);
```

```
    if n==0:
```

```
        return 1
```

```
    else:
```

```
        return n*Factorial(n-1)
```

Function Call

```
Number=int(input("Enter any positive integer:"))
```

```
print(f' The factorial of {Number} is {Factorial(Number)}')
```

Output:

Enter any positive integer: 6

The factorial of 6 is 720

9c) Create anonymous functions

Algorithm:

Step1: Start

Step 2: Create a anonymous functions using lambda keyword

Step 3: Call the anonymous function to perform specific task

Step 4: Stop

Program

```
x = lambda a, b: a + b  
print(" The variable type is",type(x))  
print(f" The sum is {x(7,8)}")
```

Output:

The variable type is <class 'function'>

The sum is 15

10 a) Write a program to demonstrate modules.**ALGORITHM:**

STEP1: Create a python script cal.py

STEP2: Create a python file input.py and import specified function defined in cal.py file

STEP3: Create a python file sample.py and import all function defined in cal.py file

PROGRAM**cal.py**

```
def summation(a,b):  
    return a+b  
def subtraction(a,b):  
    return a-b;  
def multiplication(a,b):  
    return a*b;  
def divide(a,b):  
    return a/b;
```

input.py

```
from cal import summation ,divide  
a = int(input("Enter the first number"))  
b = int(input("Enter the second number"))  
print("Sum = ",summation(a,b))  
print("Division = ",divide(a,b))
```

sample.py

```
import cal

a = int(input("Enter the first number"))
b = int(input("Enter the second number"))

print("Sum = ",cal.summation(a,b))

print("Product = ",cal.multiplication(a,b))

print("Difference = ",cal.subtract(a,b))
```

9b) Write a program to demonstrate math and random modules.

ALGORITHM:

STEP1: import math , random module

STEP2: write a code to demonstrate various inbuilt function of math and random module

PROGRAM

(a) **math module**

```
import math

print("Euler Value is :",math.e)

print("PI Value is :",math.pi)

print("Factorial :",math.factorial(5))

print("GCD :",math.gcd(5,57))

print("Square root :",math.sqrt(5))

print("Ceil Value :",math.ceil(5.35))

print("Floor Value :",math.floor(5.35))

print("Sin Theta Value:",math.sin(45))

print("Cos Theta Value:",math.cos(45))

print("Tan Theta Value:",math.tan(45))
```


(b) random module

```
import random
print("Random integers between 0 to 5:",random.randint(0,5))
print("Random numbers using randrange() between 3 to
9:",random.randrange(3,9))
print("Random Floating numbers between 0 to 1:",random.random())
list1= [1,4,True,800,"python",27,"hello"]
print("Random element from list is :",random.choice(list1))
print("Multiple random elements from the list are:",random.choices(list1,k=2))
```

11. Write a program to illustrate the numpy module.

Algorithm:

Step1: Import numpy module

Step2: Create two numpy arrays of each (3,4) shape

Step3: Perform the various arithmetic operations on arrays using built-in methods from numpy module

Step4: Use transpose() to find the transpose of the array

Step 5: Use ndim attribute to know the number of dimensions of array

Step6: Use shape attribute to know the shape of array

Step7: Use dtype attribute to know the data type of array

Step 8: Use the itemsize attribute to know the size of each item in bytes

Step 9: Use the size attribute to know the number of items in the array

Program:

```
import numpy as np
array1=np.arange(1,13).reshape(3,4)
print(" The elements of first arrays:\n",array1)
List=range(1,13)
array2=np.asarray(List).reshape(3,4)
print(" The elements of second arrays:\n",array2)
print("The  addition of two numpy arrays:")
print(np.add(array1,array2))
print("The  difference of two numpy arrays:")
```

```
print(np.subtract(array1,array2))
print("The product  of two numpy arrays:")
print(np.multiply(array1,array2))
print("The  division of two numpy arrays:")
print(np.divide(array1,array2))
print("The transpose of first matrix is:\n")
print(np.transpose(array1))
print(" Illustration of numpy array attributes:")
print("The dimension of array1:", array1.ndim)
print("The shape of array1 is", array1.shape)
print("The data type of array1 is:", array1.dtype)
print("The size of each item in the array1 is:",array1.itemsize)
print("The total number of elements in array1:", array1.size)
```

Output:

The elements of first arrays:

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

The elements of second arrays:

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

The addition of two numpy arrays:

```
[[ 2  4  6  8]
 [10 12 14 16]
 [18 20 22 24]]
```

The difference of two numpy arrays:

```
[[0 0 0 0]
 [0 0 0 0]
 [0 0 0 0]]
```

The product of two numpy arrays:

```
[[ 1  4  9 16]
 [25 36 49 64]
 [81 100 121 144]]
```

The division of two numpy arrays:

```
[[1. 1. 1. 1.]
 [1. 1. 1. 1.]
 [1. 1. 1. 1.]]
```

The transpose of first matrix is:

```
[[ 1  5  9]
 [ 2  6 10]
 [ 3  7 11]
 [ 4  8 12]]
```

Illustration of numpy array attributes:

The dimension of array1: 2

The shape of array1 is (3, 4)

The data type of array1 is: int32

The size of each item in the array1 is: 4

The total number of elements in array1: 12

12. Read the data from file and count the number of vowels and consonants present in it. Read the filename as input from the user.

Algorithm:

Step1: Start

Step2: Read the filename input from the user

Step3: Open the file in read mode

Step4: Create set vowels containing the vowels

Step5: Create set Cons containing the consonants

Step 6: Read the all the contents of file using read()

Step 7: Initialize countV to zero

Step 8: Check for each character in text. If the character in the text is
Vowel then increment the countV by 1

Step 9: Initialize countC to zero

Step 10: Check for each character in text. If the character in the text is
Consonant then increment the countC by 1

Step11: Display countC,countV

Step12: Stop

```
fileName = input("Enter the file to check: ")
```

```
infile = open(fileName, "r")
```

```
vowels = set("A E I O U a e i o u")
```

```
cons = set("b c d f g h j k l m n p q r s t v w x y z B C D F G H J K L  
M N P Q R S T V W X Y Z")
```

```
text = infile.read()
```

```
countV = 0
```

```
for V in text:
```

```
    if V in vowels:
```

```
        countV += 1
```

```
countC = 0
for C in text:
    if C in cons:
        countC += 1

print("The number of Vowels is: ",countV)
print("The number of consonants is: ",countC)
```

Output:

Enter the file to check: sample3.txt

The number of Vowels is: 7

The number of consonants is: 8