|  |  |
| --- | --- |
| **EX.NO:** 01  **DATE:** | **ADDITION, SUBTRACTION, MULTIPLICATION AND DIVISION** |

1. **Write a program for addition, subtraction, multiplication, and division of two numbers.**

## AIM:

To write a program for addition, subtraction, multiplication, and division of two numbers.

## ALGORITHM:

**STEP 1:** Display choice of operation in screen, “Choice 1: Add,

2: Sub, 3: Div, 4: Mul, 5: Quit”.

**STEP 2:** Receive input choice from the user.

**STEP 3:** Convert the input to integer (num = int (n1)).

**STEP 4:** If choice of option is 1 then receive input for A and B, Initiate operation c = A + B.

**STEP 5:** If choice of option is 2 then receive input for A and B, Initiate operation c = A - B.

**STEP 6:** If choice of option is 3 then receive input for A and B, Initiate Operation c = A / B.

**STEP 7:** If choice of option is 3 then receive input for A and B, Initiate operation c = A \* B.

## PROGRAM:

print("Choice 1: Add, 2: Sub, 3: Div, 4: Mul, 5: Quit") n = int(input("Enter the choice of operation :"))

a = int(input("Enter the value for AA :"))

b = int(input("Enter the value for B :")) if(n == 1):

print("You have chosen addition Option :") c = a + b

print("The Result",a,"+",b,"=",c) elif(n == 2):

print("You have chosen subtraction Option :") c = a - b

print("The Result",a,"-",b,"=",c) elif(n == 3):

print("You have chosen division Option :") c = a / b

print("The Result",a,"/",b,"=",c) elif(n == 4):

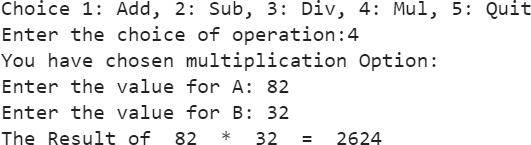
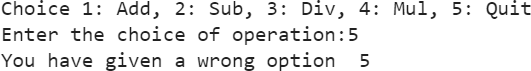
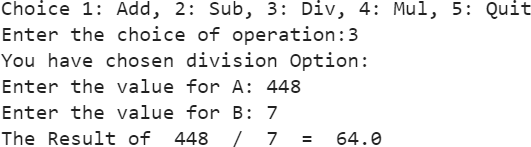
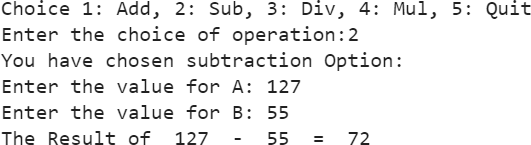
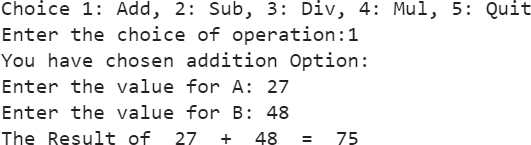
print("You have chosen multiplication Option :") c = a \* b

print ("The Result",a,"\*",b,"=",c)

elif(n==5):

print("You have given a wrong option", n) exit ()

## OUTPUT:



**RESULT:**

Thus, the program has been executed successfully and verified.

|  |  |
| --- | --- |
| **EX.NO:** 02 (i)  **DATE:** | **FIBONACCI SERIES** |

1. **i) Write a program to print Fibonacci number series AIM:**

To write a program to print Fibonacci number series.

## ALGORITHM:

**STEP 1:** Display "Fibonacci Series"

**STEP 2:** Prompt the user to enter the number (i.e., 1 to n, any integer number).

**STEP 3:** Convert the input n from string to integer (num = int(n1)).

**STEP 4:** If the received input (num) is greater than zero, then Initialize f = 0, s = 1, I = 0, and next = 0.

**STEP 5:** While the value of I is in the range of 0 to n+1, do Check whether i <= 1:

If yes, then

Print the value of I (i.e., 0).

If i is not less than 1, then

Calculate the next number as the sum of f and s. Swap the values for f and s (i.e., f = s, s = next).

Print the value in the next number.

Increment I by 1.

**STEP 6:** Break.

**STEP 7:** If the received input is not an integer, then Display "You have entered the wrong input".

## PROGRAM:

print("Fibonacci Series")

n = int(input("Enter the number :")) if(n>0):

f = 0

s = 1

i = 0

next = 0 while(1):

print("\n\n\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*") for i in range (n+1):

if (i<=1): print(i)

else:

next = f + s f = s

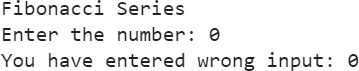
s = next print (next)

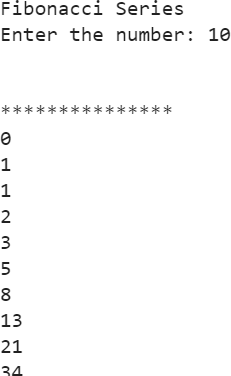
i = i + 1 break

else:

print("You have entered wrong input", n)

## OUTPUT:





**RESULT:**

Thus, the program has been executed successfully and verified.

|  |  |
| --- | --- |
| **EX.NO:** 2 (ii)  **DATE:** | **FIZZ BUZZ PROGRAM** |

**2.ii) Write a program to incorporate FIZZ for any number divisible by 3 and BUZZ for any number divisible for 5 and FIZZBUZZ for any number divisible by 3 and 5 as well.**

## AIM:

To write a program to incorporate FIZZ for any number divisible by 3 and BUZZ for any number divisible for 5 and FIZZBUZZ for any number divisible by 3 and 5 as well.

## ALGORITHM:

**STEP 1:** Display "Fizz Buzz Program:" to prompt the user.

**STEP 2:** Prompt the user to enter a number and store it in the variable n'.

**STEP 3:** Convert 'n' to an integer and store it in the variable 'num'.

**STEP 4:** Initialize a loop variable 'i' and iterate through numbers from 0 to 'num'.

**STEP 5:** Inside the loop, check if 'i' is divisible by both 3 & 5: If yes, print 'i = Fizz Buzz'.

If not, check if 'i' is divisible by 3: If yes, print 'i = Fizz'.

If not, check if 'i' is divisible by 5: If yes, print 'i = Buzz'.

If none of the conditions are met, print the current value of 'i'.

**STEP 6**: End the loop.

## PROGRAM:

print("Fizz Buzz Program :")

n = int(input("Enter the number :")) i = 1

for i in range (n+1):

if (i % 3 == 0 and i % 5 == 0): print (str(i) + "= Fizz Buzz")

elif (i % 3 == 0):

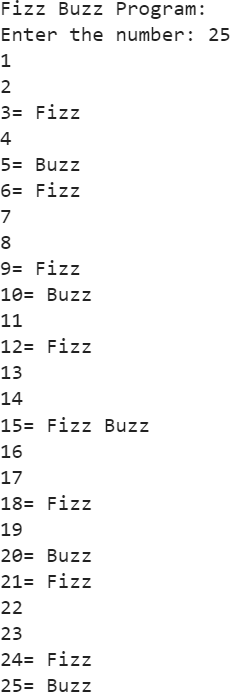
print (str(i) + "= Fizz ") elif (i % 5 == 0):

print(str(i) +"= Buzz ")

else:

print(i)

## OUTPUT:



**RESULT:**

Thus, the program has been executed successfully and verified.

|  |  |
| --- | --- |
| **EX.NO:** 03  **DATE:** | **CROWD COMPUTING** |

1. **Write a program to collect approximate cost for a material or object and store the same in the array. Remove first and last 10% of the listed cost from the array and compute the mean value of the array items.**

## AIM:

To write a program to collect approximate costs for a

material or object, store them in an array, remove the first and last 10% of the listed costs, and compute the mean value of the remaining array items.

## ALGORITHM:

**STEP 1:** Import the 'mean' function from the 'statistics' module.

**STEP 2:** Print “CROWD COMPUTING”

**STEP 3:** Create an array name “Elements” and store list of items.

**STEP 4:** Print the values stored in array elements

**STEP 5:** Use the library statistics and import mean “from statistics import mean”

**STEP 6:** Remove first and last 10 % of the listed cost from the array

**STEP 7:** Print the values stored in array elements after removal of the first and last 10% of the listed values.

**STEP 8:** Calculated the mean value of the array items.

## PROGRAM:

from statistics import mean print("CROWD COMPUTING")

print("\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*")

Estimates = [200, 100 ,250, 375 ,300 ,500 ,800 ,900 ,200 ,100 ,600,

90,250,300,350,363,397,450,500,700,275,125,125,185,225,240,31

0,415,300,250,300,2000,2500]

i = 0

for i in range (len(Estimates)): print (Estimates[i])

Estimates.sort()

print ("SORTED VALUES")

for i in range (len(Estimates)): print (Estimates[i])

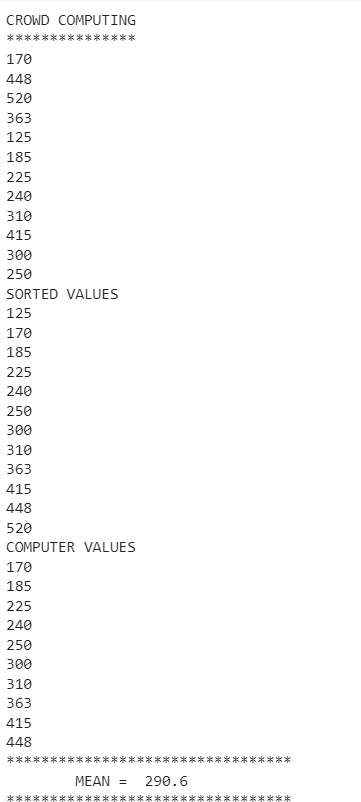
tv = int (0.1 \* len(Estimates))

Estimates = Estimates [tv:len(Estimates)-tv] print ("COMPUTER VALUES")

for i in range (len(Estimates)): print (Estimates[i])

print ("\t", mean(Estimates))

## OUTPUT:



**RESULT:**

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 04  **DATE:** | **JUMBLED WORDS** |

1. **Write a program to create a play game called jumbled word. AIM:**

To write a program to create a play game called jumbled word.

## ALGORITHM:

**STEP 1:** Define a function named ‘choose’ to randomly select

a word from a predefined list.

**STEP 2:** Define a function named ‘jumble’ to shuffle the

characters of a word and create a jumbled version.

**STEP 3:** Define a function named ‘thank’ to print the scores

and a thank-you message at the end of the game.

**STEP 4:** Define a function named ‘play’ to implement the

game logic.

**STEP 5:** Inside the ‘play’ function, initialize player names,

scores and a turn counter.

**STEP 6:** Use a while loop to continue the game until a player decides to quit.

**STEP 7:** Inside the loop, randomly select a word using the ‘choose’ function and create a jumbled version using the ‘jumble’ function.

**STEP 8:** Alternate turns between Player 1 and Player 2.

**STEP 9:** Prompt the current player to enter their guess and check if it matches the original word.

**STEP10:** Update the player’s score and provide feedback

based on the correctness of the guess.

## PROGRAM:

import random def choose ():

words = ['rainbow','computer','water','ice','type','light','zebra','arab']

pick = random.choice(words) return pick

def jumble (word):

jumbled = "".join(random.sample(word,len(word))) return jumbled

def thank (p1n,p2n,p1,p2):

print (p1n, " Your Score is : ", p1) print (p2n,"Your Score is :", p2) print ("Thanks for Playing ")

def play():

p1name = input ("Enter the name of Player 1 :") p2name = input ("Enter the name of Player 2 :") pp1 = 0

pp2 = 0

turn = 0

while (1):

picked\_word = choose() qn = jumble(picked\_word)

print ("\n\n\t\t The question is :",qn) #player 2

if (turn % 2 == 0):

print ("\n\n\t\t",p2name, "Your Turn") ans = input ("Enter the answer")

if (ans == picked\_word): pp2 = pp2 + 1

print ("The Question was :", qn)

print ("The Answer you Given : ", ans)

print ("You have given the right answer ! congratulation") else:

print ("Wrong answer ! Better Luck Next Time")

c = int (input (" Press 1 to continue or 0 to quit")) if (c == 0):

thank(p1name,p2name,pp1,pp2) break

else:

print("\n\n\t\t",p1name, "Your Turn")

ans = input("Enter the answer") if (ans == picked\_word):

pp1 = pp1 + 1

print(" The Question was : ", picked\_word) print(" The Answer you Given : ", ans) print(" You have given the right answer !")

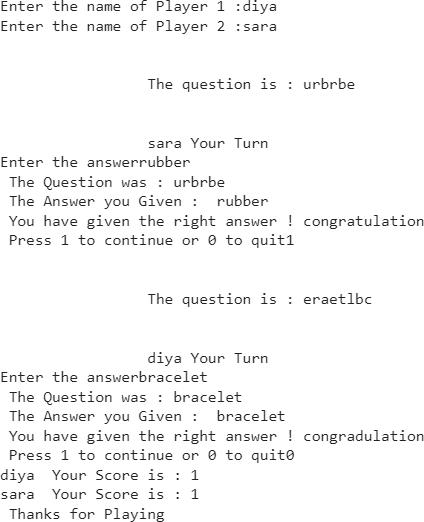
else:

print(" You have given the wrong answer !") c = input("Press 1 to continue or 0 to quit") if( c == 0):

thank(p1name,p2name,pp1,pp2) break

turn = turn + 1 play()

## OUTPUT:



**RESULT:**

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 05  **DATE:** | **BIRTHDAY PARADOX** |

1. **Write a program to random generate 50 birth dates and find how many of have same day of the year.**

## AIM:

To write a program to random generate 50 birth dates and find how many of have same day of the year.

## ALGORITHM:

**STEP 1:** Create an array variable birthday without any array elements.

**STEP 2:** Create a while loop to execute 50 times and generate year,month,day.

**STEP 3:** Create a variable dd and store year,month and day usingdatetime.date function.

**STEP 4:** Convert the birthdate stored in the dd variable to day of the year using dd.timetuple().tm\_yday and store it in the variable day\_of\_year

**STEP 5:** Append the day\_of\_year into the array birthday [] (i.e. birthday.append [day\_of\_year])

**STEP 6:** Similarly generate another 49 values for Variable day\_of\_year and append the same in the array birthday.

**STEP 7:** Sort the array elements using sort function (i.e. birthday.sort())

**STEP 8:** Print all the elements in the birthday array. You can easy notice elements having same day of the year.

## PROGRAM:

import random import datetime birthday=[]

i=0 while(i<5):

year=random.randint(1875,2019)

if (year % 4 == 0 and year % 100 == 0 or year % 400 == 0):

leap = 1 else:

leap = 0

month = random.randint(1,12) if (month==2 and leap ==1):

day = random.randint(1,29) print(day)

elif (month == 2 and leap == 0): day = random.randint(1,29) print(day)

elif (month == 7 and month == 8): day = random.randint(1,31)

elif (month %2 != 0 and month < 7): day = random.randint(1,31)

elif (month %2 != 0 and month < 7 and month < 12): day = random.randint(1,31)

else:

day = random.randint(1,30)

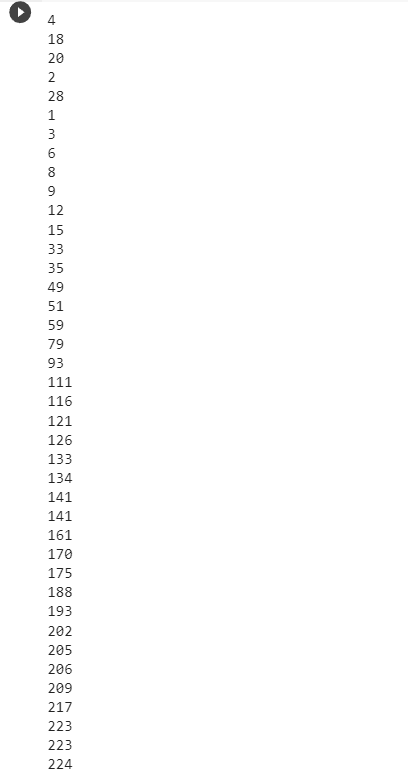
dd = datetime.date(year,month,day) day\_of\_year = dd.timetuple().tm\_yday

i = i+1 birthday.append(day\_of\_year) birthday.sort()

i=0 while(i<50):

print(birthday[i]) i = i + 1

## OUTPUT:



**RESULT:**

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 06  **DATE:** | **LIST PROGRAMS** |

1. **Write a program for the following:**
   1. **Display of list with elements.**
   2. **Finding the range of the lists.**
   3. **Indexing in the lists (including negative indexing)**
   4. **Use of loop in the lists.**
   5. **Adding, removing and joining two lists.**

## AIM:

To Write a python program to implement List Programs

* + - Display of List with elements.
    - Finding the range of the Lists.
    - Indexing in the Lists (Including Negative Indexing).
    - Use of Loop in the Lists.
    - Adding, removing and Joining two Lists

## ALGORITHM:

**STEP 1:** To display a list with elements first create a list with elements i.e, my\_list, to display the list simply print(my\_list).

**STEP 2:** To find the range, calculate its range by subtracting the minimum value from maximum value.

**STEP 3:** For indexing use index starting from 0 to access elements. Negative indexing access elements from the end of the list.

**STEP 4:** Use a for loop for iterating throughout the list.

**STEP 5:** To add an element to the end of the list, use the append() method. To remove an element by value use remove() & by index use the del keyword by list & index. To join two lists use extend() method or the ‘+’ operator to concatenate them into new list.

## PROGRAM:

my\_list = [1, 2, 3, 4, 5] print("Original list:", my\_list)

**# Calculate the range of the list** list\_range = max(my\_list) - min(my\_list) print("Range:", list\_range)

**# Access the element at index 2**

element = my\_list[2]

print ("Element at index 2:", element)

**# Access the last element**

element = my\_list[-1]

print ("Last element:", element) print ("\nLoop through the list:") for item in my\_list:

print(item)

**# Append 4 to the end of the list**

my\_list.append(4)

print("\nAfter appending 4:", my\_list) **# Remove the first occurrence of 3** my\_list.remove(3)

print("After removing the first occurrence of 3:", my\_list)

**# Remove the element at index 1**

del my\_list[1]

print("After deleting element at index 1:", my\_list)

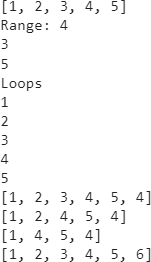
**# Combine list1 and list2 using extend**

list1 = [1, 2, 3]

list2 = [4, 5, 6] list1.extend(list2)

print("\nCombined list1 and list2:", list1)

## OUTPUT:



**RESULT:**

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 07  **DATE:** | **TUPLE PROGRAMS** |

1. **Write a program for the following:**
   1. **Creation of tuple with values.**
   2. **Finding the range of the tuple.**
   3. **Indexing in the tuple(including negative indexing)**
   4. **Use of loop in the tuple.**
   5. **Adding, removing and joining two tuple.**

## AIM:

To Write a python program to implement Tuple Programs

* Display of Tuple with elements.
* Finding the range of the Tuple.
* Indexing in the Tuple (Including Negative Indexing).
* Use of Loop in the Tuple.
* Adding, removing and Joining two Tuple

## ALGORITHM:

**STEP 1:** To create a tuple, define a tuple by enclosing values in parenthesis.

**STEP 2:** Calculate the range of tuple by subtracting the min value & max value.

**STEP 3:** For indexing use index starting from 0 to access elements. Negative indexing access elements from the end of the list.

**STEP 4:** Use a for loop for iterating throughout the tuple’s

elements.

**STEP 5:** Tuples are immutable, which means we cannot add or remove elements to an existing tuple.

**STEP 6:** To join two lists use extend() method or the ‘+’

operator to concatenate them into new list.

## PROGRAM:

my\_tuple = (10, 20, 30, 40, 50) print("Original tuple:", my\_tuple) **# Print a range**

print("Range (index 1 to 4):", my\_tuple[1:5])

**# Access the element**

element = my\_tuple[2] print("Element at index 2:", element) **# Access the last element**

element = my\_tuple[-1] print("Last element:", element)

**# Loop through the tuple and print each item**

print("\nLoop through the tuple:") for item in my\_tuple:

print(item)

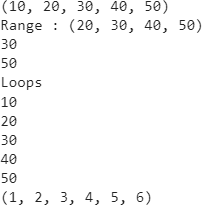
**# Combine two tuples**

tuple1 = (1, 2, 3)

tuple2 = (4, 5, 6)

combined\_tuple = tuple1 + tuple2 print("\nCombined tuple:", combined\_tuple)

## OUTPUT:



**RESULT:**

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 08  **DATE:** | **DICTIONARY PROGRAMS** |

1. **Write a program for the following:**
   1. **Display of unordered elements.**
   2. **Accessing the elements in the dictionary.**
   3. **Use of loop in the dictionary.**
   4. **Adding,removing and joining two dictionary.**

## AIM:

To Write a python program to implement Dictionary

Programs.

* + - Display of unordered elements.
    - Accessing the elements in the dictionary.
    - Use of Loop in the Dictionary.
    - Adding, removing and Joining two Dictionary

## ALGORITHM:

**STEP 1:** Create a dictionary with key value pairs. Dictionaries in python are unordered.

**STEP 2:** Access elements in a dictionary by specifying the key enclosed in square brackets.

**STEP 3:** Use a for loop for iterating throughout the keys or key value pairs in the dictionary.

**STEP 4:** To add a new key value pair, simply assign a value to the new key.

**STEP 5:** To remove a key value pair from a dictionary, use the pop() method or del keyword.

**STEP 6:** To combine two dictionaries, create a new dictionary and update it with the key\_value pairs from both the dictionaries using update() method.

## PROGRAM:

my\_dict = {'name': 'John', 'age': 30, 'city': 'New York'} print("Original dictionary:", my\_dict)

**# Accessing the value of the key** name = my\_dict['name'] print("Name:", name)

**# Loop through the dictionary and print keys**

print("\nKeys in the dictionary:") for key in my\_dict:

print(key)

**# Adding a new key-value pair**

my\_dict['gender'] = 'Male'

print("\nAfter adding gender:", my\_dict)

**# Removing the key-value pair**

my\_dict.pop('age')

print("After removing age:", my\_dict)

**# Combining two dictionaries**

dict1 = {'name': 'Alice', 'age': 25}

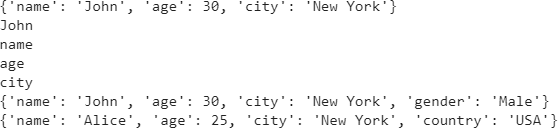
dict2 = {'city': 'New York', 'country': 'USA'}

**# Creating a new dictionary and combining them** combined\_dict = dict1.copy() # Create a copy of dict1 combined\_dict.update(dict2) # Update it with dict2

**# Display the combined dictionary**

print("\nCombined dictionary:", combined\_dict)

## OUTPUT:



**RESULT:**

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 09  **DATE:** | **SPEECH TO TEXT** |

1. **Write a python program to convert speech to text. AIM:**

To write a python program to convert speech to text.

# ALGORITHM:

**STEP 1:** Install Speech Recognition API of Google using anaconda command prompt type the command “pip install SpeechRecognition”.

**STEP 2:** Import speech recognition library as sr.

**STEP 3:** Create a audio file with WAV extension and assign the audio file to a source variable.

**STEP 4:** Initialize recognizer.

**STEP 5:** Read the audio file using the sr.AudioFile.

**STEP 6:** Print the audio file using print statement.

# PROGRAM:

import speech\_recognition as sr AUDIO\_FILE = ("harvard.wav") #use audio file as source

r=sr.Recognizer() # initialize Recognizer with sr.AudioFile(AUDIO\_FILE) as source:

audio=r.record(source) #reads the audio file

try:

print("audio file contains" +r.recognize\_google(audio)) except sr.UnknownValueError:

print (" Google Speech Recognition not understood the file uploaded")

except sr.RequestError:

print ("Couldn't get the result from Google Recognition")

# OUTPUT:

audio file contains the stale smell of old beer lingers it takes heat to bring out the odor a cold dip restores health and zest a salt pickle taste fine with ham tacos al pastor are my favorite a zestful food is the hot cross bun

## RESULT:

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 10  **DATE:** | **MONTE HALL\_3 – DOORS AND A TWIST** |

1. **Write a program to create a game “Monte Hall\_3 – Doors and a Twist.**

## AIM:

To write a program to create a game “Monte Hall\_3 – Doors and a Twist”. This comprises of three doors. In which two doors contain GOAT and one door contain BMW. User has to pick his/her choice of door. If the choice of door contains BMW then user WINS otherwise LOST.

## ALGORITHM:

**STEP 1:** Importing random library as we are using random choice in the program

**STEP 2:** Create an array doors[0] \* 3(Initializing the array as doors[0]=0, doors[1]=0,doors[2]=0)

**STEP 3:** Create an array goatdoors=[]

**STEP 4:** Initializing a variable swap=0, don’t\_swap=0 & j=0

**STEP 5:** Create a while loop to execute the loop instructions for 10 times

**STEP 6:** Declaring a variable x and initializing with random number from 0 to 2

**STEP 7:** Passing the value of x as an array item number to array variable doors[] and store “BMW” i.e., doors[0 or 1 or 2]=”BMW”

**STEP 8:** Create a for loop with variable i to execute three times (for i in range (0,3)) and check if i equals to x then continue, otherwise store “Goat” in array doors[] (i.e., doors[i]=”Goat”)

**STEP 9:** Accept user choice of input to the variable choice, user input choice to be 0 or 1 or 2

**STEP 10:** Open a door that comprises of goat by creating a variable door\_open and apply random.choice function to select a random item from array goatdoor[] and store it in door\_open variable (i.e., door\_open = random.choice(goatdoor))

**STEP 11:** Accept user choice of swap to the variable ch as either YorN

**STEP 12:** If user choice of swap is yes (i.e., ch==’Y’), then check user given choice door contain “Goat”. If user door choice contains “Goat” then declare user “Player Wins”. Increment swap variable with 1

**STEP 13:** If user choice for swap is no (i.e., ch==’N’), then check user given choice door contain “Goat”. If user door choice contains “Goat” then declare user “Player Wins”. Increment swap variable with 1

**STEP 14:** Increment j=j+1

**STEP 15:** Print “no of swap wins” and “no of don’t swap wins”

## PROGRAM:

import random

doors=[0]\*3, goatdoor=[]\*2, swap=0 dont\_swap=0, j=0

while(j<10):

x=random.randint(0,2) doors[x]="BMW"

for i in range (0,3): if (i==x):

continue else:

doors[i]="Goat" goatdoor.append(i)

choice=int(input("Enter your choice 0,1,2 : " )) door\_open=random.choice(goatdoor) while(door\_open==choice):

door\_open=random.choice(goatdoor)

ch = input("Do you want to swap? y/n: ") if(ch=='y'):

if(doors[choice]=='Goat'): print ("Players wins") swap=swap+1

else:

print("player lost")

else:

if (doors[choice]=='Goat'): print("Player lost")

else:

print("Player wins") dont\_swap=dont\_swap+1 j=j+1

print("No of swap wins",swap)

print("No of dont swap wins",dont\_swap)

## OUTPUT:

Enter your choice 0,1,2 : 1 Do you want to swap? y/n: y player lost

Enter your choice 0,1,2 : 0 Do you want to swap? y/n: y player lost

Enter your choice 0,1,2 : 1 Do you want to swap? y/n: n Player lost

Enter your choice 0,1,2 : 1 Do you want to swap? y/n: y Players wins

## RESULT:

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 11  **DATE:** | **PLOTS** |

1. **Write a program to plot values in chart with x-axis and y-axis. AIM:**

To write a program to plot values in chart with x-axis and y-axis.

## ALGORITHM:

**STEP 1:** Import plotting library <matplotlib.pyplot> as plt

**STEP 2:** Print “\* PLOTTING VALUES IN CHART WITH X-AXIS AND Y-AXIS \*”

**STEP 3:** Plot values X-axis (1,2,3,4) and Y-axis (10,13,20,25) with blue Square symbol

## PROGRAM:

import matplotlib.pyplot as plt print(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”) print(“\* \*”)

print(“\* PLOTTING VALUES IN CHART WITH X-AXIS AND Y-AXIS\*”)

print(“\* \*”)

print(“\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*”) plt.plot([1,2,3,4],[10,13,20,25],’bs’)

## OUTPUT:

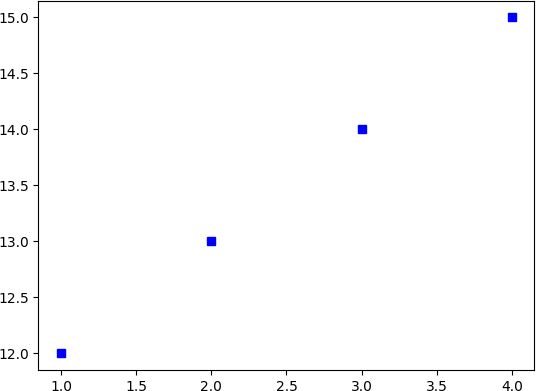
\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

\* \*

\* PLOTTING VALUES IN CHART WITH X-AXIS AND Y-AXIS \*

\* \*

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* [<matplotlib.lines.Line2D at 0x7acf9038e6e0>]



## RESULT:

Thus, the program has been successfully executed and verified.

|  |  |
| --- | --- |
| **EX.NO:** 12  **DATE:** | **PANDAS DATAFRAME** |

1. **Write a python program using pandas library to perform the following operation.**

**AIM:** To write a python program using pandas library to perform the following operations.

* Create Data Frame
* Manipulate the values in Data Frame
* Bar Charts
* Pie Charts
* Scatter Plots

## ALGORITHM:

**STEP 1:** Import the Pandas library and Matplotlib for plotting

**STEP 2:** Create a dictionary with data for the Data Frame,

including ‘Age’, ’Weight’, and ‘Blood Pressure’.

**STEP 3:** Manipulation the values in the data frame (split the Blood Pressure column into two columns and Calculate BMI)

**STEP 4:** Display the modified Data Frame.

**STEP 5:** Generate a bar chart for BMI, a pie chart for Systolic Blood Pressure, and a scatter plot for BMI vs Systolic Blood Pressure GDP growth.

**STEP 6:** Display the plots using plt.show()

## PROGRAM:

import pandas as pd

import matplotlib.pyplot as plt

**# Create a DataFrame**

data = {'Patient': ['John Doe', 'Jane Doe', 'Peter Parker', 'Mary Jane Watson', 'Bruce Wayne'],

'Age': [35, 25, 20, 18, 45],

'Weight': [80, 60, 70, 55, 90],

'Systolic Blood Pressure': [120, 110, 130, 100, 140],

'Diastolic Blood Pressure': [80, 70, 90, 60, 100]} df = pd.DataFrame(data)

**# Display the Original DataFrame** print("Original DataFrame:") print(df)

print("\n")

**# Split the Blood Pressure column into two columns**

df['Systolic Blood Pressure'] = pd.to\_numeric(df['Systolic Blood Pressure'])

df['Diastolic Blood Pressure'] = pd.to\_numeric(df['Diastolic Blood Pressure'])

**# Calculate BMI**

df['Height'] = 1.75 # Assume all patients are 1.75 meters tall df['BMI'] = df['Weight'] / (df['Height'] \* df['Height'])

**# Display the modified DataFrame** print("Modified DataFrame:") print(df)

print("\n")

**# Create a bar chart for BMI**

df.plot(kind='bar', x='Patient', y='BMI', title='BMI Bar Chart') plt.xlabel('Patient')

plt.ylabel('BMI') plt.show()

**# Create a pie chart for Systolic Blood Pressure**

df.plot.pie(y='Systolic Blood Pressure', labels=df['Patient'], autopct='%1.1f%%',

title='Systolic Blood Pressure Pie Chart') plt.show()

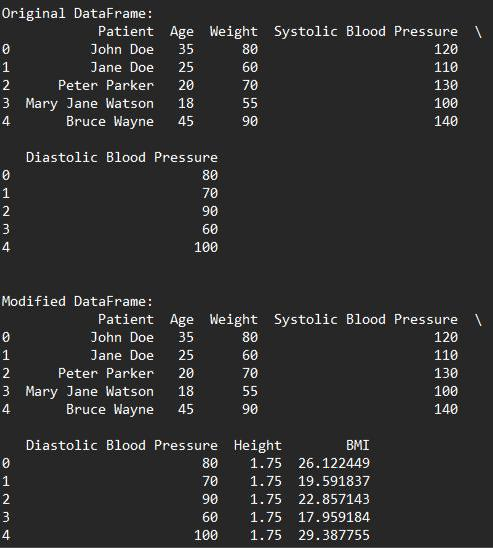
**# Create a scatter plot for BMI vs Systolic Blood Pressure**

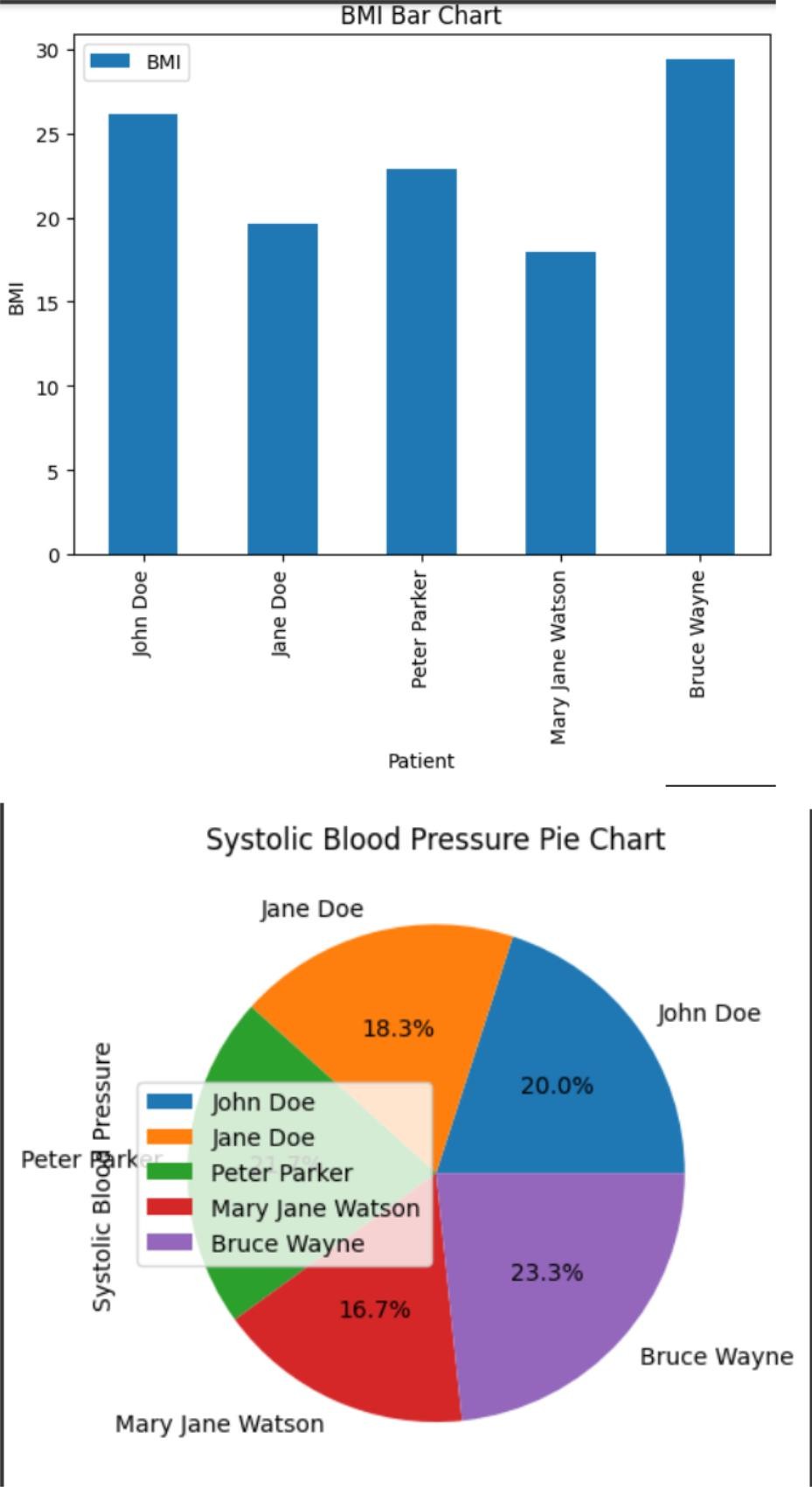
df.plot(kind='scatter', x='BMI', y='Systolic Blood Pressure', title='BMI vs Systolic Blood Pressure Scatter Plot')

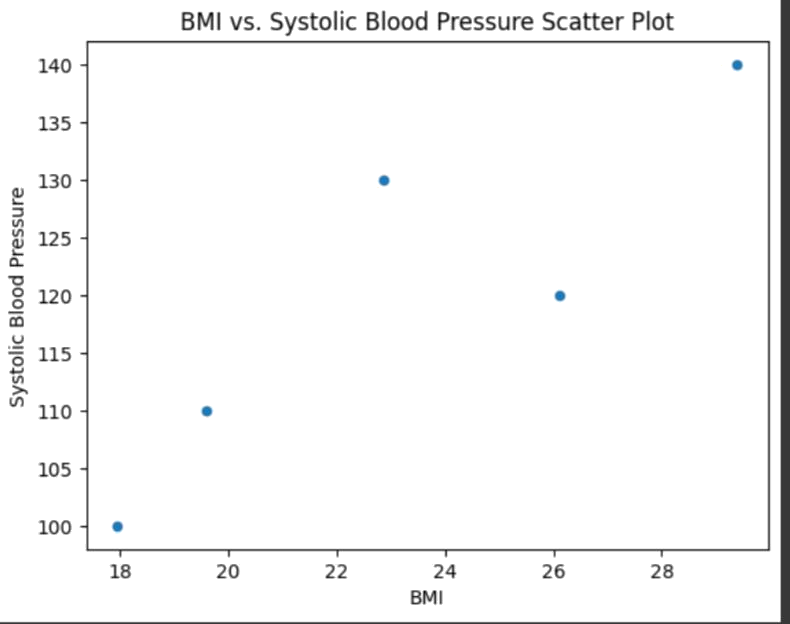
plt.xlabel('BMI')

plt.ylabel('Systolic Blood Pressure') plt.show()

# OUTPUT:







**RESULT:**

Thus, the program has been successfully executed and verified.