**Python**

**2. LOOPS**

**21 .**

**get a list a = [2,3,4,55,33,4,55,343,66,77,88,99] and write a program to browse over this list and print previous number and current number**

**Code:**

**a = [2, 3, 4, 55, 33, 4, 55, 343, 66, 77, 88, 99]**

**for i in range(1, len(a)):**

**previous\_number = a[i - 1]**

**current\_number = a[i]**

**print(f"Previous Number: {previous\_number}, Current Number: {current\_number}")**

**O/P:**

**Previous Number: 2, Current Number: 3**

**Previous Number: 3, Current Number: 4**

**Previous Number: 4, Current Number: 55**

**Previous Number: 55, Current Number: 33**

**Previous Number: 33, Current Number: 4**

**Previous Number: 4, Current Number: 55**

**Previous Number: 55, Current Number: 343**

**Previous Number: 343, Current Number: 66**

**Previous Number: 66, Current Number: 77**

**Previous Number: 77, Current Number: 88**

**Previous Number: 88, Current Number: 99**

**22.**

**Get a number from user, write a program to calculate factorial of the number using for loop and while loop**

**Code:**

**num=int(input("enter the number: "))**

**fact=1;**

**for i in range(1,num+1):**

**fact=fact\*i**

**print("factorial value: ",fact)**

**O/P:**

**enter the number: 5**

**factorial value: 120**

**23.**

**Get number from user and reverse number in other variable and print output**

**Code:**

**number = int(input("Enter the integer number: "))**

**revs\_number = 0**

**while (number > 0):**

**remainder = number % 10**

**revs\_number = (revs\_number \* 10) + remainder**

**number = number // 10**

**print("The reverse number is : {}".format(revs\_number))**

**O/P:**

**Enter the integer number: 123**

**The reverse number is : 321**

**24.**

**print patterns 1 right angeled**

**Code:**

**n = 5**

**for i in range(0, n):**

**for j in range(0, i+1):**

**print("\*", end=" ")**

**print(" ")**

**O/P:**

**\***

**\* \***

**\* \* \***

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

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

**25.**

**print patterns 2 Pyramid**

**Code:**

**num\_rows = int(input("Enter the number of rows " ));**

**for i in range(0, num\_rows):**

**for j in range(0, num\_rows-i-1):**

**print(end=" ")**

**for j in range(0, i+1):**

**print("\*", end=" ")**

**print()**

**O/P:**

**Enter the number of rows 5**

**\***

**\* \***

**\* \* \***

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

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

**26.**

**print pattern upside down pyramid**

**Code:**

**n = 5**

**for i in range(n, 0, -1):**

**for j in range(n - i):**

**print(" ", end="")**

**for k in range(2\*i - 1):**

**print("\*", end="")**

**print("\r")**

**O/P:**

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

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

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

**\*\*\***

**\***

**27.**

**print patterns --4 (A) right angeled**

**Code:**

**n = 5**

**for i in range(0, n):**

**for j in range(0, i+1):**

**print("A", end=" ")**

**print(" ")**

**O/P:**

**A**

**A A**

**A A A**

**A A A A**

**A A A A A**

**28.**

**print pattern 5(#) reverse**

**n = 5**

**for i in range(n, 0, -1):**

**for j in range(n - i):**

**print(" ", end="")**

**for k in range(2\*i - 1):**

**print("#", end="")**

**print("\r")**

**#########**

**#######**

**#####**

**###**

**#**

**29.**

**print pattern 5(#) right Angeled**

**Code:**

**n = 5**

**for i in range(0, n):**

**for j in range(0, i+1):**

**print("#", end=" ")**

**print(" ")**

**O/P:**

**#**

**# #**

**# # #**

**# # # #**

**# # # # #**

**30.**

**print patterns -- NUMBERS right angeled**

**Code:**

**n = 5**

**num = 1**

**for i in range(0, n):**

**num = 1**

**for j in range(0, i + 1):**

**print(num, end=" ")**

**num = num + 1**

**print(" ")**

**O/P:**

**1**

**1 2**

**1 2 3**

**1 2 3 4**

**1 2 3 4 5**

**31.**

**print patterns -- NUMBERS sequence right angeled**

**Code:**

**n = 5**

**num = 1**

**for i in range(0, n):**

**for j in range(0, i + 1):**

**print(num, end=" ")**

**num = num + 1**

**print(" ")**

**O/P:**

**1**

**2 3**

**4 5 6**

**7 8 9 10**

**11 12 13 14 15**

**32.**

**ask user to pick any number between 0 and 100**

**and implement a guessing game program to identify number picked up by user.**

**(use while loop)**

**Code:**

**target\_number = 76**

**attempts = 0**

**game\_won = False**

**print("Welcome to the Number Guessing Game!")**

**while not game\_won:**

**user\_guess = int(input("Guess the number between 0 and 100: "))**

**attempts =attempts+1**

**if user\_guess < target\_number:**

**print("Too low! Try again.")**

**elif user\_guess > target\_number:**

**print("Too high! Try again.")**

**else:**

**print(f"Congratulations! You guessed the number {target\_number} in {attempts} attempts.")**

**game\_won = True**

**print("Thanks for playing!")**

**O/P:**

**Welcome to the Number Guessing Game!**

**Guess the number between 0 and 100: 34**

**Too low! Try again.**

**Guess the number between 0 and 100: 77**

**Too high! Try again.**

**Guess the number between 0 and 100: 76**

**Congratulations! You guessed the number 76 in 3 attempts.**

**Thanks for playing!**

**33. import random**

**def display\_matchsticks(matchsticks):**

**print("Matchsticks remaining:", matchsticks)**

**print("| " \* matchsticks)**

**print()**

**def user\_move():**

**while True:**

**try:**

**move = int(input("Your move (1-4 matchsticks): "))**

**if 1 <= move <= 4:**

**return move**

**else:**

**print("Invalid move. Please choose between 1 and 4 matchsticks.")**

**except ValueError:**

**print("Invalid input. Please enter a number.")**

**def computer\_move(matchsticks):**

**# Simple computer strategy: Always leave a multiple of 4 matchsticks.**

**number = random.randint(1, 4)**

**matchsticks=number**

**return (matchsticks)**

**matchsticks = 21**

**while matchsticks > 0:**

**display\_matchsticks(matchsticks)**

**# User's move**

**user\_matches = user\_move()**

**matchsticks -= user\_matches**

**if matchsticks <= 0:**

**print("Congratulations! You win!")**

**break**

**display\_matchsticks(matchsticks)**

**# Computer's move**

**computer\_matches = computer\_move(matchsticks)**

**print("Computer's move:", computer\_matches)**

**matchsticks -= computer\_matches**

**if matchsticks <= 0:**

**print("Oops! You lose. Better luck next time.")**

**break**

**O/P:**

**Matchsticks remaining: 21**

**| | | | | | | | | | | | | | | | | | | | |**

**Your move (1-4 matchsticks): 1**

**(GAMEPLAY)**

**34.**

**Get a number from user and check if number is twin prime number**

**Code:**

**def is\_prime(num):**

**i = 2**

**count = 0**

**while i < num:**

**if (num % i == 0):**

**count = 1**

**break**

**i += 1**

**if count == 1:**

**return False**

**else:**

**print("prime")**

**return True**

**def is\_twin\_prime(num):**

**if is\_prime(num):**

**return (is\_prime(num - 2) or is\_prime(num + 2))**

**return False**

**num = int(input("Enter a number: "))**

**if is\_twin\_prime(num):**

**print(f"{num} is a twin prime number.")**

**else:**

**print(f"{num} is not a twin prime number.")**

**O/P:**

**Enter a number: 3**

**prime**

**prime**

**3 is a twin prime number.**

**35.**

**get a number from user and check number of occurances of a single digit in that number.**

**(for example. Num=7888, check number of occurances of digit 8 in number)**

**Code:**

**user\_number = int(input("Enter a number: "))**

**digit\_to\_count = int(input("Enter the digit to count: "))**

**number\_str = str(user\_number)**

**occurrences = 0**

**for digit in number\_str:**

**if int(digit) == digit\_to\_count:**

**occurrences += 1**

**print(f"The digit {digit\_to\_count} appears {occurrences} time(s) in the number {user\_number}.")**

**O/P:**

**Enter a number: 788**

**Enter the digit to count: 8**

**The digit 8 appears 2 time(s) in the number 788.**

**36.**

**get a number from user and check if the number is perfect square.**

**Code:**

**num=int(input("enter the number: "))**

**sqrt\_num = num \*\* 0.5**

**if sqrt\_num.is\_integer():**

**print("The provided number is a perfect square")**

**else:**

**print("The provided number is not a perfect square")**

**O/P:**

**enter the number: 4**

**The provided number is a perfect square**

**37.**

**Read product file and tran\_dtl file and implement inner join using loops (use two for loops)**

**Code:**

**with open(r"C:\Users\HP\Desktop\Philomath\SQL\Retail\_Data\product.csv", 'r') as product\_file:**

**product\_data = [line.strip().split(',') for line in product\_file]**

**with open(r"C:\Users\HP\Desktop\Philomath\SQL\Retail\_Data\tran\_dtl\_1\_2019.csv", 'r') as tran\_dtl\_file:**

**tran\_dtl\_data = [line.strip().split(',') for line in tran\_dtl\_file]**

**inner\_join\_result = []**

**for product\_row in product\_data:**

**for tran\_dtl\_row in tran\_dtl\_data:**

**if product\_row[0] == tran\_dtl\_row[1]:**

**inner\_join\_result.append(product\_row + tran\_dtl\_row)**

**for row in inner\_join\_result:**

**print(row)**

**O/P:**

**['99', 'Roasted Pistachios', '4.99', 'Snacks', '3', '2020-04-15T19:57:01.676996\_343', '99', '3', '14.97', '01-27-2019']**

**['99', 'Roasted Pistachios', '4.99', 'Snacks', '3', '2020-04-15T19:57:01.677995\_383', '99', '1', '4.99', '01-30-2019']**

**38.**

**Read product file and tran\_dtl file and implement inner join using loops (use two for loops)**

**-- implement left, right and full outer join**

**Code:**

**with open(r"C:\Users\HP\Desktop\Philomath\SQL\Retail\_Data\product.csv", 'r') as product\_file:**

**product\_data = [line.strip().split(',') for line in product\_file]**

**with open(r"C:\Users\HP\Desktop\Philomath\SQL\Retail\_Data\tran\_dtl\_1\_2019.csv", 'r') as tran\_dtl\_file:**

**tran\_dtl\_data = [line.strip().split(',') for line in tran\_dtl\_file]**

**# inner join**

**inner\_join\_result = []**

**for product\_row in product\_data:**

**for tran\_dtl\_row in tran\_dtl\_data:**

**if product\_row[0] == tran\_dtl\_row[1]:**

**inner\_join\_result.append(product\_row + tran\_dtl\_row)**

**for row in inner\_join\_result:**

**print(row)**

**# left join**

**left\_join\_result = []**

**print("Left Join : ")**

**for product\_row in product\_data:**

**for tran\_dtl\_row in tran\_dtl\_data:**

**if product\_row[0] == tran\_dtl\_row[1]:**

**left\_join\_result.append(product\_row + tran\_dtl\_row )**

**left\_join\_result.append("INN joined")**

**else:**

**left\_join\_result.append(product\_row)**

**left\_join\_result.append("LEFT remaining")**

**for row in left\_join\_result:**

**print(row)**

**# right join**

**right\_join\_result = []**

**print("Right Join : ")**

**for product\_row in product\_data:**

**for tran\_dtl\_row in tran\_dtl\_data:**

**if product\_row[0] != tran\_dtl\_row[1]:**

**right\_join\_result.append(tran\_dtl\_row)**

**right\_join\_result.append("right remaining")**

**else:**

**right\_join\_result.append(product\_row + tran\_dtl\_row)**

**right\_join\_result.append("INN joined")**

**for row in right\_join\_result:**

**print(row)**

**# outer join**

**outer\_join\_result = []**

**print("Outer Join : ")**

**for product\_row in product\_data:**

**for tran\_dtl\_row in tran\_dtl\_data:**

**if product\_row[0] == tran\_dtl\_row[1]:**

**outer\_join\_result.append(product\_row + tran\_dtl\_row)**

**outer\_join\_result.append("in")**

**else:**

**outer\_join\_result.append(tran\_dtl\_row)**

**outer\_join\_result.append(product\_row)**

**outer\_join\_result.append("OUT")**

**for row in outer\_join\_result:**

**print(row)**

**O/P:**

**All respective joins.**