1.WRITE A PROGRAM TO DETERMINE OPERATOR PROCEDURE TO EVALUATE AN EXPRESSION

a=10

a+=5

print("a+=5:",a)

a-=5

print("a-=5:",a)

a\*=5

print("a\*=5:",a)

a/=5

print("a/=5:",a)

a%=3

print("a%=3:",a)

a\*\*=3

print("a\*\*=3:",a)

a//=3

print("a//=3",a)

OUTPUT:

a+=5: 15

a-=5: 10

a\*=5: 50

a/=5: 10.0

a%=3: 1.0

a\*\*=3: 1.0

a//=3 0.0

2.WRITE A PROGRAM TO DEMONSTRATE OPERATOR PRECEDENCE

AND EVALUATE AN EXPRESSION Programp = 20

q = 10

r = 15

s = 5

t = 0

# Calculate (p+q)\*r/s

t = (p + q) \* r / s

print("Value of (p+q)\*r/s is", t)

# Calculate (p+q)\*(r/s)

t = (p + q) \* (r / s)

print("Value of (p+q)\*(r/s) is", t)

# Calculate ((p+q)\*r)/s

t = ((p + q) \* r) / s

print("Value of ((p+q)\*r)/s is", t)

# Calculate p + (q\*r)/s

t = p + (q \* r) / s

print("Value of p+(q\*r)/s is",t)

OUTPUT:

value of (p+q)\*r/s is 90.0 value of (p+q)\*(r/s) is 90.0 value of ((p+q)\*r)/s is 90.0 value of p+(q\*r)/s is 50.

03.WRITE A PROGRAM TO DISPLAY THE GRADE OF A STUDENT

USING ELIF STATEMENT Program:

print("Enter 'x' for exit")

print("Enter marks obtained in five subjects:")

# Input first mark

mark1 = input()

if mark1.lower() == 'x':

exit()

else:

mark1 = int(mark1)

# Input the remaining marks

mark2 = int(input("Enter marks for subject 2: "))

mark3 = int(input("Enter marks for subject 3: "))

mark4 = int(input("Enter marks for subject 4: "))

mark5 = int(input("Enter marks for subject 5: "))

# Calculate the sum and average

total = mark1 + mark2 + mark3 + mark4 + mark5

average = total / 5

# Determine and print the grade

if 91 <= average <= 100:

print("YOUR GRADE IS A+")

elif 81 <= average <= 90:

print("YOUR GRADE IS A")

elif 71 <= average <= 80:

print("YOUR GRADE IS B+")

elif 61 <= average <= 70:

print("YOUR GRADE IS B")

elif 51 <= average <= 60:

print("YOUR GRADE IS C+")

elif 41 <= average <= 50:

print("YOUR GRADE IS C")

elif 0 <= average <= 40:

print("YOUR GRADE IS F")

else:

print("STRANGE GRADE…..!!")

OUTPUT:

Enter ‘x’ for exit

Enter marks obtained in 5 subjects

65

79

89

97

76

Your grade is A

4.WRITE A PROGRAM TO IMPLEMENT FLOYD TRIANGLE

USING FOR LOOP Program :

print("Enter 'x' to exit")

ran = input("Upto how many lines? ")

if ran == 'x':

exit()

else:

rang = int(ran)

k = 1

for i in range(1, rang + 1):

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

print(k, end="")

k += 1

print()

OUTPUT:

Enter ‘x’ to exit

Upto how many lines?3

1

2 3 4

5 6 7 8

5.WRITE A PROGRAM TO CHECK WHETHER THE GIVEN NUMBER IS PRIME OR NOT USING FOR LOOP

Program:

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

if num > 1:

for i in range(2, num):

if num % i == 0:

print(num, "is not a prime number")

print(i, "times", num // i, "is", num)

break

else:

print(num, "is a prime number")

else:

print(num, "is not a positive number")

OUTPUT:

Enter a number 57

57 is a prime number

6. WRITE A PROGRAM TO COMPUTE THE GCD OF NUMBER USING FUNCTIONS

Program:

def computegcd(x, y):

if x < y:

smaller = x

else:

smaller = y

for i in range(1, smaller + 1):

if (x % i == 0) and (y % i == 0):

gcd = i

return gcd

num1 = int(input("Enter a number 1: "))

num2 = int(input("Enter a number 2: "))

print("GCD of", num1, "and", num2, "is", computegcd(num1, num2))

OUTPUT:

Enter a number 1: 4

Enter a number 2: 12

GCD of 4 and 12 is 4

7. FINDING FACTORIAL OF A NUMBER USING RECURSION FUNCTION Program:

def recur\_factorial(n):

if n == 1:

return n

else:

return n \* recur\_factorial(n - 1)

num = 7

if num < 0:

print("Sorry, the factorial does not exist for negative numbers")

elif num == 0:

print("The factorial of 0 is 1")

else:

print("The factorial of", num, "is", recur\_factorial(num))

Output:

The factorial of 7 is 5040

8.TAKE A LIST OF WORDS AND RETURN THE LENGTH OF THE

LONGEST ONE USING STRING Program: a = []

n = int(input("Enter the number of elements in the list: "))

for x in range(0, n):

element = input("Enter element " + str(x + 1) + ": ")

a.append(element)

max1 = len(a[0])

temp = a[0]

for i in a:

if len(i) > max1:

max1 = len(i)

temp = i

print("The word with the longest length is:", temp)

OUTPUT:

Enter the number of elements in list:3

Enter element 1:apple

Enter element 2:mango

Enter element 3:pineapple

The word with longest length is pineapple

9(A).WRITE A PROGRAM TO PERFORM LINEAR SEARCH USING

STRINGS

Programs:

list\_of\_string = ['apple', 'orange', 'grape', 'mango']

x = input("Enter the string to be searched: ")

found = False

for i in range(len(list\_of\_string)):

if list\_of\_string[i].lower() == x.lower():

found = True

print(f"{x} found at {i + 1}th position")

break

if not found:

print(f"{x} is not in the list")

def binary\_search(item\_list, item):

first = 0

last = len(item\_list) - 1

found = False

while first <= last and not found:

mid = (first + last) // 2

if item\_list[mid] == item:

found = True

elif item < item\_list[mid]:

last = mid - 1

else:

first = mid + 1

return found

a = [10, 20, 30, 40, 50]

print(a)

b = int(input("Enter the number to be searched: "))

if binary\_search(a, b):

print(f"{b} is found in the list")

else:

print(f"{b} is not in the list")

OUTPUT:

[10,20,30,40,50]

Enter the number to searched:10

True

10.WRITE A PROGRAM TO IMPLEMENTATION LIST AS ARRAYS

[MULTIPLYINGMATRIS]

Program:

x = [[12, 7, 3], [4, 5, 6], [7, 8, 9]]

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

result = [[0, 0, 0], [0, 0, 0], [0, 0, 0]]

for i in range(len(x)):

for j in range(len(y[0])):

for k in range(len(y)):

result[i][j] += x[i][k] \* y[k][j]

for r in result:

print(r)

OUTPUT:

[61,83,105]

[66,81,96]

[105,90,150]

11.WRITE A PROGRAM DEMONSTRATE USE OF LIST AND RELATED FUNCTIONS

Program:

my\_list = []

my\_list = [1, 2, 3]

my\_list = [1, "hello", 3, 4]

my\_list = ["mouse", [8, 4, 6], [8]]

my\_list = ['p', 'r', 'o', 'b', 'e']

print(my\_list[0])

print(my\_list[2])

print(my\_list[4])

n\_list = ["happy", [2, 0, 1, 5]]

print(n\_list[0][1])

print(n\_list[1][3])

my\_list = ['p', 'r', 'o', 'b', 'e']

print(my\_list[-1])

print(my\_list[-5])

my\_list = ['p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z']

print(my\_list[2:5])

print(my\_list[:-5])

print(my\_list[5:7])

print(my\_list[:])

odd = [2, 4, 6, 8]

odd[0] = 1

print(odd)

odd = [1, 3, 5]

odd.append(7)

print(odd)

odd.extend([9, 11, 13])

print(odd)

odd = [1, 3, 5]

print(odd + [9, 7, 5])

print(["re"] \* 3)

my\_list = ["p", "r", "o", "b", "i", "e", "m"]

del my\_list[2]

print(my\_list)

del my\_list[1:5]

print(my\_list)

my\_list = [3, 8, 1, 6, 0, 8, 4]

print(my\_list.index(8))

print(my\_list.count(8))

my\_list.sort()

print(my\_list)

my\_list.reverse()

print(my\_list)

pow2 = [2 \*\* x for x in range(10)]

print(pow2)

for fruit in ['apple', 'banana', 'mango']:

print("I like", fruit)

Output;

p

o

e

a

5

e

p

['o', 'g', 'r']

['p', 'r', 'o', 'g']

['a', 'm']

['p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z']

[1, 4, 6, 8]

[1, 3, 5, 7]

[1, 3, 5, 7, 9, 11, 13]

[1, 3, 5, 9, 7, 5]

['re', 're', 're']

['p', 'r', 'b', 'i', 'e', 'm']

['p', 'm']

1

2

[0, 1, 3, 4, 6, 8, 8]

[8, 8, 6, 4, 3, 1, 0]

[1, 2, 4, 8, 16, 32, 64, 128, 256, 512]

I like apple

I like banana

I like mango

12.WRITE A PROGRAM TO DEMONSTRATE USE OF

TUPLE,SET AND RELATED FUNCTION

Program

my\_tuple = ()

my\_tuple = (1, 2, 3)

print(my\_tuple)

my\_tuple = (1, "Hello", 3, 4)

print(my\_tuple)

my\_tuple = ('Mouse', [8, 4, 6], (1, 2, 3))

print(my\_tuple)

my\_tuple = 3, 4, 6, "Dog"

print(my\_tuple)

my\_tuple = ('p', 'e', 'r', 'm', 'i', 't')

print(my\_tuple[0])

print(my\_tuple[5])

n\_tuple = ('Mouse', [8, 4, 6], (1, 2, 3))

my\_tuple = ('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z')

print(my\_tuple[1:4])

print(my\_tuple[:-7])

print(my\_tuple[7:])

print(my\_tuple[:])

my\_tuple = (4, 2, 3, [6, 5])

my\_tuple[3][0] = 9

print(my\_tuple)

my\_tuple = ('a', 'p', 'p', 'l', 'e')

print(my\_tuple.count('p'))

print(my\_tuple.count('l'))

print('a' in my\_tuple)

print('b' in my\_tuple)

for name in ('John', 'Kate'):

print("Hello", name)

E = {0, 2, 4, 6, 8}

N = {1, 2, 3, 4, 5}

print("Union of E and N is", E | N)

print("Intersection of E and N is", E & N)

print("Difference of E and N is", E - N)

print("Symmetric difference of E and N is", E ^ N)

Output:

(1, 2, 3)

(1, 'Hello', 3, 4)

('Mouse', [8, 4, 6], (1, 2, 3))

(3, 4, 6, 'Dog') p

t

('r', 'o', 'g')

('p', 'r')

('i', 'z')

('p', 'r', 'o', 'g', 'r', 'a', 'm', 'i', 'z')

(4, 2, 3, [9, 5])

2

1

True

False

Hello John

Hello Kate

Union of E and N is {0, 1, 2, 3, 4, 5, 6, 8}

Intersection of E and N is {2, 4}

Difference of E and N is {0, 6, 8}

Symmetric difference of E and N is {0, 1, 3, 5, 6, 8}

13.TO DEMONSRATE THE USE OF DICTIONARY AND

RELATED FUNCTIONS

Program:

my\_dict = {}

my\_dict = {1: "apple", 2: "ball"}

my\_dict = {1: 'apple', 2: 'ball'}

my\_dict = {'name': 'Jack', 'age': 24}

print(my\_dict['name'])

print(my\_dict.get('age'))

my\_dict = {'name': 'Jack', 'age': 26}

my\_dict['age'] = 27

print(my\_dict)

my\_dict['address'] = 'downtown'

print(my\_dict)

squares = {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

print(squares.pop(4))

print(squares)

print(squares.popitem())

print(squares)

del squares[3]

print(squares)

squares.clear()

print(squares)

squares = {1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

print(squares.items())

print(sorted(squares))

marks = {}.fromkeys(['maths', 'english', 'science'], 0)

print(marks)

for item in marks.items():

print(item)

odd\_squares = {x: x \* x for x in range(11) if x % 2 == 1}

print(odd\_squares)

squares = {x: x \* x for x in range(6)}

print(squares)

squares = {1: 1, 3: 9, 9: 81, 5: 25, 7: 49}

print(1 in squares)

print(2 not in squares)

print(49 in squares)

for key in squares:

print(squares[key])

print(len(squares))

Output:

Jack

24

{'name': 'Jack', 'age': 27}

{'name': 'Jack', 'age': 27, 'address': 'downtown'}

16

{1: 1, 2: 4, 3: 9, 5: 25}

(5, 25)

{1: 1, 2: 4, 3: 9}

{1: 1, 2: 4}

{}

dict\_items([(1, 1), (2, 4), (3, 9), (4, 16), (5, 25)])

[1, 2, 3, 4, 5]

{'maths': 0, 'english': 0, 'science': 0}

('maths', 0)

('english', 0)

('science', 0)

{1: 1, 3: 9, 5: 25, 7: 49, 9: 81}

{0: 0, 1: 1, 2: 4, 3: 9, 4: 16, 5: 25}

True

True

False

1

9

81

25

49 5

14.WRITE A PROGRAM THAT TAKES COMMAND LINE

ARGUMENT[WORD COUNT]

Program:

import re import string frequency=() document\_text=open(‘sam.text','r') text\_string=document\_text.read().lower() match\_pattern=re.findall(r'\b[a-z]{3,16}\b',text\_string) for words in match\_pattern: count=frequency.get(words,0) frequency[words]=count+1 frequency\_list=frequency.keys() for words in frequency\_list: print(words,frequency[words]) TEXT FILE:

Maths

Chemistry

Maths

Python

OUTPUT:

(Maths,2)

(Chemistry,1)

(python,1)

15.PROGRAM THAT COMMAND LINE ARGUMENTS(WORD COUNT)

Program:

fname=input("Enter a file name:")

num\_words=0 with open(fname,'r') as f: for line in f: words=line.split() num\_words+=len(words) print("Number of words:") print(num\_words) TEXT FILE:

Maths

Python

Physics

Chemistry OUTPUT:

Enter a file name:test.txt

Number of words:4

Pet Keeper:

first is to install game module via command prompt by using the command

pip install pygame

then, we can create the game with some basic stock picture from online

Program:

from tkinter import HIDDEN, NORMAL, Tk, Canvas

def toggle\_eyes():

    current\_color = c.itemcget(eye\_left, 'fill')

    new\_color = c.body\_color if current\_color == 'white' else 'white'

    current\_state = c.itemcget(pupil\_left, 'state')

    new\_state = NORMAL if current\_state == HIDDEN else HIDDEN

    c.itemconfigure(pupil\_left, state=new\_state)

    c.itemconfigure(pupil\_right, state=new\_state)

    c.itemconfigure(eye\_left, fill=new\_color)

    c.itemconfigure(eye\_right, fill=new\_color)

def blink():

    toggle\_eyes()

    root.after(250, toggle\_eyes)

    root.after(3000, blink)

def toggle\_pupils():

    if not c.eyes\_crossed:

        c.move(pupil\_left, 10, -5)

        c.move(pupil\_right, -10, -5)

        c.eyes\_crossed = True

    else:

        c.move(pupil\_left, -10, 5)

        c.move(pupil\_right, 10, 5)

        c.eyes\_crossed = False

def toggle\_tongue():

    if not c.tongue\_out:

        c.itemconfigure(tongue\_tip, state=NORMAL)

        c.itemconfigure(tongue\_main, state=NORMAL)

        c.tongue\_out = True

    else:

        c.itemconfigure(tongue\_tip, state=HIDDEN)

        c.itemconfigure(tongue\_main, state=HIDDEN)

        c.tongue\_out = False

def cheeky(event):

    toggle\_tongue()

    toggle\_pupils()

    hide\_happy(event)

    root.after(1000, toggle\_tongue)

    root.after(1000, toggle\_pupils)

    return

def show\_happy(event):

    if (20 <= event.x and event.x <= 350) and (20 <= event.y and event.y <= 350):

        c.itemconfigure(cheek\_left, state=NORMAL)

        c.itemconfigure(cheek\_right, state=NORMAL)

        c.itemconfigure(mouth\_happy, state=NORMAL)

        c.itemconfigure(mouth\_normal, state=HIDDEN)

        c.itemconfigure(mouth\_sad, state=HIDDEN)

        c.happy\_level = 10

    return

def hide\_happy(event):

    c.itemconfigure(cheek\_left, state=HIDDEN)

    c.itemconfigure(cheek\_right, state=HIDDEN)

    c.itemconfigure(mouth\_happy, state=HIDDEN)

    c.itemconfigure(mouth\_normal, state=NORMAL)

    c.itemconfigure(mouth\_sad, state=HIDDEN)

    return

def sad():

    if c.happy\_level == 0:

        c.itemconfigure(mouth\_happy, state=HIDDEN)

        c.itemconfigure(mouth\_normal, state=HIDDEN)

        c.itemconfigure(mouth\_sad, state=NORMAL)

    else:

        c.happy\_level -= 1

    root.after(5000, sad)

root = Tk()

root.title("Screen pet")

c = Canvas(root, width=400, height=400)

c.configure(bg='dark blue', highlightthickness=0)

c.body\_color = 'SkyBlue1'

body = c.create\_oval(35, 20, 365, 350, outline=c.body\_color, fill=c.body\_color)

ear\_left = c.create\_polygon(75, 80, 75, 10, 165, 70, outline=c.body\_color, fill=c.body\_color)

ear\_right = c.create\_polygon(255, 45, 325, 10, 320, 70, outline=c.body\_color, fill=c.body\_color)

foot\_left = c.create\_oval(65, 320, 145, 360, outline=c.body\_color, fill=c.body\_color)

foot\_right = c.create\_oval(250, 320, 330, 360, outline=c.body\_color, fill=c.body\_color)

eye\_left = c.create\_oval(130, 110, 160, 170, outline='black', fill='white')

pupil\_left = c.create\_oval(140, 145, 150, 155, outline='black', fill='black')

eye\_right = c.create\_oval(230, 110, 260, 170, outline='black', fill='white')

pupil\_right = c.create\_oval(240, 145, 250, 155, outline='black', fill='black')

mouth\_normal = c.create\_line(170, 250, 200, 272, 230, 250, smooth=1, width=2, state=NORMAL)

mouth\_happy = c.create\_line(170, 250, 200, 282, 230, 250, smooth=1, width=2, state=HIDDEN)

mouth\_sad = c.create\_line(170, 250, 200, 232, 230, 250, smooth=1, width=2, state=HIDDEN)

tongue\_main = c.create\_rectangle(170, 250, 230, 290, outline='red', fill='red', state=HIDDEN)

tongue\_tip = c.create\_oval(170, 285, 230, 300, outline='red', fill='red', state=HIDDEN)

cheek\_left = c.create\_oval(70, 180, 120, 230, outline='pink', fill='pink', state=HIDDEN)

cheek\_right = c.create\_oval(280, 180, 330, 230, outline='pink', fill='pink', state=HIDDEN)

c.pack()

c.bind('<Motion>', show\_happy)

c.bind('<Leave>', hide\_happy)

c.bind('<Double-1>', cheeky)

c.happy\_level = 10

c.eyes\_crossed = False

c.tongue\_out = False

root.after(1000, blink)

root.after(5000, sad)

root.mainloop()

OUTPUT:



