

vsvc-calculator-3

May 23, 2024

```
[76]: import math
def programmer():
    def decimal_to_binary(decimal):
        return bin(decimal)[2:]
    def decimal_to_octal(decimal):
        return oct(decimal)[2:]
    def decimal_to_hex(decimal):
        return hex(decimal).upper()[2:]
    def binary_to_decimal(binary):
        return int(binary, 2)
    def binary_to_hex(binary):
        decimal = binary_to_decimal(binary)
        return decimal_to_hex(decimal)
    def binary_to_octal(binary):
        decimal = binary_to_decimal(binary)
        return decimal_to_octal(decimal)
    def hex_to_decimal(hexadecimal):
        return int(hexadecimal, 16)
    def hex_to_binary(hexadecimal):
        decimal = hex_to_decimal(hexadecimal)
        return decimal_to_binary(decimal)
    def hex_to_octal(hexadecimal):
        decimal = hex_to_decimal(hexadecimal)
        return decimal_to_octal(decimal)
    def octal_to_decimal(octal):
        return int(octal, 8)
    def octal_to_binary(octal):
        decimal = octal_to_decimal(octal)
        return decimal_to_binary(decimal)
    def octal_to_hex(octal):
        decimal = octal_to_decimal(octal)
        return decimal_to_hex(decimal)
    def is_binary(num):
        for i in num:
            if i not in "01":
                return False
        return True
```

```

def is_octal(num):
    for i in num:
        if i not in "01234567":
            return False
    return True
def is_decimal(num):
    for i in num:
        if i not in "0123456789":
            return False
    return True
def is_hex(num):
    for i in num:
        if i not in "0123456789ABCDEF":
            return False
    return True
def add_binary(num1,num2):
    return bin(int(num1, 2) + int(num2, 2))[2:]
def sub_binary(num1, num2):
    return bin(int(num1, 2) - int(num2, 2))[2:]
def add_octal(num1,num2):
    return oct(int(num1, 8) + int(num2, 8))[2:]
def sub_octal(num1,num2):
    return oct(int(num1, 8) - int(num2, 8))[2:]
def add_hexadecimal(num1,num2):
    return hex(int(num1, 16) + int(num2, 16)).upper()[2:]
def sub_hexadecimal(num1,num2):
    return hex(int(num1, 16) - int(num2, 16)).upper()[2:]
def base_conversions():
    a3="y"
    while a3=="y":
        print()
        print("Select the base of the decimal system:")
        print("2->Binary")
        print("8->Octal")
        print("10->Decimal")
        print("16->Hexadecimal")
        print()
        base=int(input("Enter the base:"))
        if base==2:
            print()
            num=input("Enter the Binary number:")
            b=is_binary(num)
            if b==False:
                print("The Number Entered is Not Binary")
            else:
                print("Binary Number : ",num)
                print("Decimal : ",binary_to_decimal(num))

```

```

        print("Octal:", binary_to_octal(num))
        print("Hexadecimal:", binary_to_hex(num))
elif base==8:
    print()
    num=input("Enter the Octal number:")
    o=is_octal(num)
    if o==False:
        print("The Number Entered is Not Octal")
    else:
        print("Octal:", num)
        print("Decimal:", octal_to_decimal(num))
        print("Binary:", octal_to_binary(num))
        print("Hexadecimal:", octal_to_hex(num))
elif base==10:
    print()
    num=input("Enter the Decimal number:")
    d=is_decimal(num)
    if d==False:
        print("The Number Entered is Not Decimal")
    else:
        print("Decimal:", int(num))
        print("Binary:", decimal_to_binary(int(num)))
        print("Hexadecimal:", decimal_to_hex(int(num)))
        print("Octal:", decimal_to_octal(int(num)))
elif base==16:
    print()
    num=input("Enter the Hexadecimal number:").upper()
    h=is_hex(num)
    if h==False:
        print("The Number Entered is Not Hexadecimal")
    else:
        print("Hexadecimal:", num)
        print("Decimal:", hex_to_decimal(num))
        print("Binary:", hex_to_binary(num))
        print("Octal:", hex_to_octal(num))
else:
    print("Invalid Choice")
print()
a3=input("Do you want to Continue Base Operations \nPress 'y' to_
↪continue : ").lower()
def arith():
    a4="y"
    while a4=="y":
        print()
        print("Select the base of the decimal system:")
        print("2->Binary")
        print("8->Octal")

```

```

print("16->Hexadecimal")
print()
base=int(input("Enter the base:"))
if base==2:
    a5="y"
    while a5=="y":
        print()
        print("Select one \n1.Addition \n2.Subtraction")
        print()
        c=int(input("Enter choice : "))
        if c==1:
            print()
            num1=input("Enter the Binary number:")
            num2=input("Enter the Binary number:")
            b1=is_binary(num1)
            b2=is_binary(num2)
            if b1==False or b2==False:
                print()
                print("The Numbers Entered are Not Binary")
            else:
                print()
                print("Binary Addition")
                print(num1,"+",num2,"=",add_binary(num1,num2))
        elif c==2:
            print()
            num1=input("Enter the Binary number:")
            num2=input("Enter the Binary number:")
            b1=is_binary(num1)
            b2=is_binary(num2)
            if b1==False or b2==False:
                print()
                print("The Numbers Entered are Not Binary")
            else:
                print()
                print("Binary Subtraction")
                print(num1,"-",num2,"=",sub_binary(num1,num2))
        else:
            print("Invalid Choice")
        print()
        a5=input("Do you want to Continue Binary Operations \nPress_
↳ 'y' to continue : ").lower()
    elif base==8:
        a5="y"
        while a5=="y":
            print()
            print("Select one \n1.Addition \n2.Subtraction")
            print()

```

```

c=int(input("Enter choice : "))
if c==1:
    print()
    num1=input("Enter the Octal number:")
    num2=input("Enter the Octal number:")
    o1=is_octal(num1)
    o2=is_octal(num2)
    if o1==False or o2==False:
        print()
        print("The Numbers Entered are Not Octal")
    else:
        print()
        print("Octal Addition")
        print(num1,"+",num2,"=",add_octal(num1,num2))
elif c==2:
    print()
    num1=input("Enter the Octal number:")
    num2=input("Enter the Octal number:")
    o1=is_octal(num1)
    o2=is_octal(num2)
    if o1==False or o2==False:
        print()
        print("The Numbers Entered are Not Octal")
    else:
        print()
        print("Octal Subtraction")
        print(num1,"-",num2,"=",sub_octal(num1,num2))
else:
    print("Invalid Choice")
print()
a5=input("Do you want to Continue Octal Operations \nPress_
↵ 'y' to continue : ").lower()
elif base==16:
    a5="y"
    while a5=="y":
        print()
        print("Select one \n1.Addition \n2.Subtraction")
        print()
        c=int(input("Enter choice : "))
        if c==1:
            print()
            num1=input("Enter the Hexadecimal number:").upper()
            num2=input("Enter the Hexadecimal number:").upper()
            h1=is_hex(num1)
            h2=is_hex(num2)
            if h1==False or h2==False:
                print()

```

```

        print("The Numbers Entered are Not Hexadecimal")
    else:
        print()
        print("Hexadecimal Addition")
        print(num1,"+",num2,"=",add_hexadecimal(num1,num2))
elif c==2:
    print()
    num1=input("Enter the Hexadecimal number:").upper()
    num2=input("Enter the Hexadecimal number:").upper()
    h1=is_hex(num1)
    h2=is_hex(num2)
    if h1==False or h2==False:
        print()
        print("The Numbers Entered are Not Hexadecimal")
    else:
        print()
        print("Hexadecimal Subtraction")
        print(num1,"-",num2,"=",add_hexadecimal(num1,num2))
else:
    print("Invalid Choice")
    print()
    a5=input("Do you want to Continue Hexadecimal Operations_")
    ↪\nPress 'y' to continue : ").lower()
    else:
        print("Invalid Choice")
    print()
    a4=input("Do you want to Continue Arithmetic Operations \nPress 'y'")
    ↪to continue : ").lower()
def shift():
    a6="y"
    while a6=="y":
        print()
        print("Select One:")
        print("1.Right Shift")
        print("2.Left Shift")
        print("3.Exit")
        print()
        c=int(input("Enter your choice : "))
        if c==1:
            print()
            num=int(input("Enter the number: "))
            shif=int(input("Enter the number of shifts: "))
            print()
            print(num,">>",shif,"is",num >> shif)
        elif c==2:
            print()
            num=int(input("Enter the number: "))

```

```

        shif=int(input("Enter the number of shifts: "))
        print()
        print(num,"<<",shif,"is",num << shif)
    elif c==3:
        break
    else:
        print("Invalid Choice")
        print()
        a6=input("Do you want to Continue Shifting Operations \nPress 'y' to
↳to continue : ").lower()
        a2="y"
        while a2=="y":
            print()
            print("Programmer Operations")
            print()
            print("Menu")
            print("1.Base")
            print("2.Arithmetic Operations of Binary/Octal/Hexadecimal Numbers : ")
            print("3.Shifting")
            print("4.Exit")
            print()
            c=int(input("Enter your choice:"))
            if c==1:
                base_conversions()
            elif c==2:
                arith()
            elif c==3:
                shift()
            elif c==4:
                break
            else:
                print("Invalid Choice")
                print()
                a2=input("Do you want to Continue Programmer Operations \nPress 'y' to
↳continue : ").lower()
def algebra():
    import cmath
    import numpy as np
    import math
    def read(r1,r2,c1,c2,a,b,n):
        if n==1:
            print("Reading Matrix A")
            print()
            for i in range(r1):
                row=[]
                for j in range(c1):
                    row.append(int(input("Enter the element:")))

```

```

        a.append(row)
    print()
else:
    print("Reading Matrix A")
    print()
    for i in range(r1):
        row=[]
        for j in range(c1):
            row.append(int(input("Enter the element:")))
        a.append(row)
    print()
    print("Reading Matrix B")
    print()
    for i in range(r2):
        row=[]
        for j in range(c2):
            row.append(int(input("Enter the element:")))
        b.append(row)
    print()
def display(r1,r2,c1,c2,a,b,n):
    print()
    if n==1:
        print("Displaying Matrix A")
        print()
        for i in range(r1):
            for j in range(c1):
                print(a[i][j],end=" ")
            print()
        print()
    else:
        print("Displaying Matrix A")
        print()
        for i in range(r1):
            for j in range(c1):
                print(a[i][j],end=" ")
            print()
        print()
        print("Displaying Matrix B")
        print()
        for i in range(r2):
            for j in range(c2):
                print(b[i][j],end=" ")
            print()
        print()
def sum_mat(r1,c1,a,b):
    result=[]
    for i in range(r1):

```



```

        row=[]
        for j in range(c1):
            row.append(a[i][j]+b[i][j])
        result.append(row)
    print("Displaying Result Matrix ")
    print()
    for i in range(r1):
        for j in range(c1):
            print(result[i][j],end=" ")
        print()
    print()
def sub_mat(r1,c1,a,b):
    result=[]
    for i in range(r1):
        row=[]
        for j in range(c1):
            row.append(a[i][j]-b[i][j])
        result.append(row)
    print("Displaying Result Matrix ")
    print()
    for i in range(r1):
        for j in range(c1):
            print(result[i][j],end=" ")
        print()
    print()
def scalar(r1,c1,a,c):
    result=[]
    for i in range(r1):
        row=[]
        for j in range(c1):
            row.append(0)
        result.append(row)
    for i in range(r1):
        for j in range(c1):
            result[i][j]+=a[i][j]*c
    print()
    print("Product Matrix")
    print()
    for i in range(r1):
        for j in range(c1):
            print(result[i][j],end=" ")
        print()
    print()
def multi(r1,r2,c1,c2,a,b):
    result=[]
    for i in range(r1):
        row=[]

```

```

        for j in range(c2):
            row.append(0)
        result.append(row)
    for i in range(r1):
        for j in range(c2):
            for k in range(c1):
                result[i][j] += a[i][k] * b[k][j]

    print()
    print("Product Matrix")
    print()
    for i in range(r1):
        for j in range(c2):
            print(result[i][j], end=" ")
        print()
    print()

def transpose(r1, c1, t, a):
    for i in range(c1):
        row = []
        for j in range(r1):
            row.append(0)
        t.append(row)
    for i in range(c1):
        for j in range(r1):
            t[i][j] = a[j][i]

    print()
    print("Transpose Matrix")
    print()
    for i in range(c1):
        for j in range(r1):
            print(t[i][j], end=" ")
        print()
    print()

def inverse(a):
    #det=np.linalg.det(a)
    inv= np.linalg.inv(a)
    return inv

def trace_mat(r1, c1, a):
    trace=0
    for i in range(r1):
        for j in range(c1):
            if i==j:
                trace+=a[i][j]
    print("Trace of the Matrix : ", trace)

def sum_rc(r1, c1, a):
    for i in range(r1):
        row=0
        for j in range(c1):

```

```

        row+=a[i][j]
    print("Sum of row",i+1," : ",row)
for i in range(c1):
    col=0
    for j in range(r1):
        col+=a[j][i]
    print("Sum of column",i+1," : ",col)
def quad():
    a = int(input("Enter coefficient of x^2: "))
    b = int(input("Enter coefficient of x: "))
    c = int(input("Enter constant: "))
    print()
    dis = (b**2) - (4*a*c)
    if dis > 0:
        rt1 = (-b + math.sqrt(dis)) / (2*a)
        rt2 = (-b - math.sqrt(dis)) / (2*a)
        print("Roots are Real and Distinct")
        print("Roots are : ", round(rt1,4)," and ", round(rt2,4))
    elif dis == 0:
        rt = -b / (2*a)
        print("Roots are Real and Equal")
        print("Roots are : ",round(rt,4)," and ", round(rt,4))
    else:
        real = -b / (2*a)
        imag = cmath.sqrt(abs(dis)) / (2*a)
        rt1 = complex(real, imag)
        rt2 = complex(real, -imag)
        print("Roots are Imaginary")
        print("Roots are : ",rt1," and ", rt2)
def cubic():
    a=int(input("Enter coefficinet of x^3:"))
    b=int(input("Enter coefficinet of x^2:"))
    c=int(input("Enter coefficinet of x:"))
    d=int(input("Enter constant:"))
    print()
    coefficients=[a, b, c, d]
    roots = np.roots(coefficients)
    print("Roots are : ",roots)
def equation():
    an="y"
    while an=="y":
        print()
        print("Equation Solver")
        print("1.Quadrctic Equation")
        print("2.Cubic Equation")
        print("3.Exit")
        print()

```

```

e_ch=int(input("Enter your choice:"))
print()
if e_ch==1:
    quad()
elif e_ch==2:
    cubic()
elif e_ch==3:
    break
else:
    print("Invalid Choice")
print()
an=input("Do you want to continue Equation Operations \n Press 'y'␣
↳to continue : ").lower()
def matrix():
    ans="y"
    while ans=="y":
        print()
        print("Welcome to Matrix Operations")
        print()
        print("___MENU___")
        print("1.Sum of Matrices")
        print("2.Difference of Matrices")
        print("3.Matrix Multiplication")
        print("4.Determinant of Matrix")
        print("5.Transpose of Matrix")
        print("6.Inverse of Matrix")
        print("7.Trace of Matrix")
        print("8.Sum of rows and columns of Matrix")
        print("9.Exit from Matrix Operations")
        print()
        ch=int(input("Enter your choice:"))
        print()
        if ch==1:
            n=0
            a=[]
            b=[]
            r1=int(input("Enter no of rows of matrix A:"))
            c1=int(input("Enter no of columns of matrix A:"))
            r2=int(input("Enter no of rows of matrix B:"))
            c2=int(input("Enter no of columns of matrix B:"))
            print()
            if r1<=0 or c1<=0 or r2<=0 or c2<=0:
                print("Error: Enter Natural numbers")
            else:
                if r1==r2 and c1==c2:
                    read(r1,r2,c1,c2,a,b,n)
                    display(r1,r2,c1,c2,a,b,n)

```

```

        sum_mat(r1,c1,a,b)
    else:
        print("Matrix Addition Not Possible!!!!")
elif ch==2:
    n=0
    a=[]
    b=[]
    r1=int(input("Enter no of rows of matrix A:"))
    c1=int(input("Enter no of columns of matrix A:"))
    r2=int(input("Enter no of rows of matrix B:"))
    c2=int(input("Enter no of columns of matrix B:"))
    print()
    if r1<=0 or c1<=0 or r2<=0 or c2<=0:
        print("Error: Enter Natural numbers")
    else:
        if r1==r2 and c1==c2:
            read(r1,r2,c1,c2,a,b,n)
            display(r1,r2,c1,c2,a,b,n)
            sub_mat(r1,c1,a,b)
        else:
            print("Matrix Subtraction Not Possible!!!!")
elif ch==3:
    a1="y"
    while a1=="y":
        print()
        print("Select one:")
        print("1.Scalar Multiplication")
        print("2.Matrix multiplication")
        print()
        choice=int(input("Enter your choice:"))
        if choice==1:
            print()
            a=[]
            r1=int(input("Enter no of rows of matrix A:"))
            c1=int(input("Enter no of columns of matrix A:"))
            print()
            r2=0
            c2=0
            b=0
            n=1
            print()
            if r1<=0 or c1<=0:
                print("Error: Enter Natural numbers")
                print()
            else:
                read(r1,r2,c1,c2,a,b,n)
                display(r1,r2,c1,c2,a,b,n)

```

```

        c=int(input("Enter the constant to be multiplied_
↪with:"))

        scalar(r1,c1,a,c)
    elif choice==2:
        print()
        n=0
        a=[]
        b=[]
        r1=int(input("Enter no of rows of matrix A:"))
        c1=int(input("Enter no of columns of matrix A:"))
        r2=int(input("Enter no of rows of matrix B:"))
        c2=int(input("Enter no of columns of matrix B:"))
        print()
        if r1<=0 or c1<=0 or r2<=0 or c2<=0:
            print("Error: Enter Natural numbers")
        else:
            if r2==c1:
                read(r1,r2,c1,c2,a,b,n)
                display(r1,r2,c1,c2,a,b,n)
                multi(r1,r2,c1,c2,a,b)
            else:
                print("Matrix Multiplication Not Possible!!!")
        else:
            print("Invalid Choice")
            a1=input("Do you want to Continue Multiplication Operation_
↪\n Press 'y' to Continue : ").lower()
    elif ch==4:
        print()
        a=[]
        r1=int(input("Enter no of rows of matrix A:"))
        c1=int(input("Enter no of columns of matrix A:"))
        r2=0
        c2=0
        b=0
        n=1
        print()
        if r1<=0 or c1<=0:
            print("Error: Enter Natural numbers")
        else:
            if r1==c1:
                read(r1,r2,c1,c2,a,b,n)
                display(r1,r2,c1,c2,a,b,n)
                det=np.linalg.det(a)
                print("Determinant of the Matrix: ", det)
            else:
                print()

```

```

        print("Given Matrix is Not a Square Matrix.Determinant_
↳Cannot be Calculated!!")
    elif ch==5:
        print()
        n=1
        a=[]
        t=[]
        r2=0
        c2=0
        b=0
        r1=int(input("Enter no of rows of matrix A:"))
        c1=int(input("Enter no of columns of matrix A:"))
        print()
        if r1<=0 or c1<=0:
            print("Error: Enter Natural numbers")
        else:
            read(r1,r2,c1,c2,a,b,n)
            display(r1,r2,c1,c2,a,b,n)
            transpose(r1,c1,t,a)
    elif ch==6:
        print()
        n=1
        a=[]
        r2=0
        c2=0
        b=0
        r1=int(input("Enter no of rows of matrix A:"))
        c1=int(input("Enter no of columns of matrix A:"))
        print()
        if r1<=0 or c1<=0:
            print("Error: Enter Natural numbers")
        else:
            if r1==c1:
                read(r1,r2,c1,c2,a,b,n)
                display(r1,r2,c1,c2,a,b,n)
                det=np.linalg.det(a)
                if det==0:
                    print("The Matrix is Singular and doesn't have an_
↳Inverse.")
                    continue
                else:
                    x=inverse(a)
                    print("Inverse of Matrix A")
                    print(x)
            else:
                print("Given Matrix is Not a Square Matrix.Inverse_
↳Cannot be Calculated!!")

```

```

elif ch==7:
    print()
    n=1
    a=[]
    r2=0
    c2=0
    b=0
    r1=int(input("Enter no of rows of matrix A:"))
    c1=int(input("Enter no of columns of matrix A:"))
    print()
    if r1<=0 or c1<=0:
        print("Error: Enter Natural numbers")
    else:
        if r1==c1:
            read(r1,r2,c1,c2,a,b,n)
            display(r1,r2,c1,c2,a,b,n)
            trace_mat(r1,c1,a)
        else:
            print("Given Matrix is Not a Square Matrix Trace Cannot_
↳be Calculated")
elif ch==8:
    print()
    n=1
    a=[]
    r2=0
    c2=0
    b=0
    r1=int(input("Enter no of rows of matrix A:"))
    c1=int(input("Enter no of columns of matrix A:"))
    print()
    if r1<=0 or c1<=0:
        print("Error: Enter Natural numbers")
    else:
        read(r1,r2,c1,c2,a,b,n)
        display(r1,r2,c1,c2,a,b,n)
        sum_rc(r1,c1,a)
elif ch==9:
    break
else:
    print("Invalid Choice")
    print()
    ans=input("Do you want to continue to Matrix Operations \n Press_
↳'y' to continue : ").lower()
    answer="y"
    while answer=="y":
        print()
        print("Algebra Operations")

```



```

print()
print("Menu")
print("1.Matrix Operations")
print("2.Equations")
print("3.Exit")
print()
a_ch=int(input("Enter your choice:"))
if a_ch==1:
    matrix()
elif a_ch==2:
    equation()
elif a_ch==3:
    break
else:
    print("Invalid Choice")
print()
answer=input("Do you want to continue to Algebra Operations \n Press_
↪ 'y' to continue : ").lower()
def standard():
    import re

    def basic_calc():
        def calculate(expression):
            def operation(num1, num2, operator):
                if operator == '+':
                    return num1 + num2
                elif operator == '-':
                    return num1 - num2
                elif operator == '*':
                    return num1 * num2
                elif operator == '/':
                    if num2 == 0:
                        return "Error"
                    else:
                        return num1 / num2

            # Tokenize the expression
            t = re.findall(r'\d+\.\d*|([()+\-*/])', expression)

            # Operator precedence
            precedence = {'+': 1, '-': 1, '*': 2, '/': 2}

            def evaluate(expression):
                operators = []
                operands = []
                for t in expression:
                    if t.isdigit() or '.' in t:

```

```

        operands.append(float(t))
    elif t in '+-*/':
        while (operators and operators[-1] in precedence and
               precedence[operators[-1]] >= precedence[t]):
            num2 = operands.pop()
            num1 = operands.pop()
            op = operators.pop()
            operands.append(operation(num1, num2, op))
        operators.append(t)
    elif t == '(':
        operators.append(t)
    elif t == ')':
        while operators[-1] != '(':
            num2 = operands.pop()
            num1 = operands.pop()
            op = operators.pop()
            operands.append(operation(num1, num2, op))
        operators.pop()
    while operators:
        num2 = operands.pop()
        num1 = operands.pop()
        op = operators.pop()
        operands.append(operation(num1, num2, op))
    return operands[0]

return evaluate(t)

expression = input("Enter expression (terminate with =)")
while '=' not in expression:
    expression += input()

if expression[0] == '-':

    expression = '0' + expression

result = calculate(expression)
print("Result:", result)

def reciprocal():
    def recipro():
        if number == 0:
            print("Math error")
        else:
            print("1/", number, "=", 1/number)
    number=float(input("Enter number : "))
    recipro()

```

```

def square():
    def sq(number):
        return pow(number,2)
    number = float(input("Enter number : "))
    print(sq(number))

def sqroot():
    import math
    def square_root(number):
        if number == 0:
            return 0
        elif number < 0:
            print("Math ERROR")
            return None
        else:
            return math.sqrt(number)
    number = float(input("Enter number: "))
    result = square_root(number)
    if result is not None:
        print(result)

def neg():
    def negate_number(number):
        if number == 0:
            return 0
        elif number > 0:
            return -number
        else:
            return abs(number)
    number = float(input("Enter number : "))
    print(negate_number(number))

ans='y'
while ans=='y':
    print("Menu","\n","1.Basic Calculation","\n","2.Reciprocal","\n","3.
↪Square","\n","4.Square Root","\n","5.Negation","\n","6.Exit")
    ch=int(input("Enter choice : "))
    if ch==1:
        basic_calc()
    elif ch==2:
        reciprocal()
    elif ch==3:
        square()
    elif ch==4:
        sqroot()
    elif ch==5:

```

```

        neg()
    elif ch==6:
        break
    else :
        print("Invalid choice")
    ans=input("Do you wish to continue? (y/n) : ")
def scientific():
    def trigonometric_functions():
        while True:
            print("Trigonometric Functions")
            print("1. Sine")
            print("2. Cosine")
            print("3. Tangent")
            print("4. Cosecant")
            print("5. Secant")
            print("6. Cotangent")
            print("7. Exit to Scientific Calculator")
            choice = input("Enter choice: ")

            if choice == '1':
                angle = float(input("Enter angle in degrees: "))
                radian = math.radians(angle)
                result = math.sin(radian)
                print("Sine of {} degrees is {:.4f}".format(angle, result))

            elif choice == '2':
                angle = float(input("Enter angle in degrees: "))
                radian = math.radians(angle)
                result = math.cos(radian)
                print("Cosine of {} degrees is {:.4f}".format(angle, result))

            elif choice == '3':
                angle = float(input("Enter angle in degrees: "))
                radian = math.radians(angle)
                result = math.tan(radian)
                rl = [90 + i * 180 for i in range(100)]
                if angle in rl:
                    print("Tangent is undefined")
                else:
                    print("Tangent of {} degrees is {:.4f}".format(angle,
↪result))

            elif choice == '4':
                angle = float(input("Enter angle in degrees: "))
                radian = math.radians(angle)
                sine_value = math.sin(radian)
                rl = [180 + i * 180 for i in range(100)]

```

```

        if angle in r1 or angle==0:
            print("Cosecant is undefined")
        else:
            result = 1 / sine_value
            print("Cosecant of {} degrees is {:.4f}".format(angle,
↪result))

    elif choice == '5':
        angle = float(input("Enter angle in degrees: "))
        radian = math.radians(angle)
        cosine_value = math.cos(radian)
        r1 = [90 + i * 180 for i in range(100)]
        if angle in r1:
            print("Secant is undefined")
        else:
            result = 1 / cosine_value
            print("Secant of {} degrees is {:.4f}".format(angle,
↪result))

    elif choice == '6':
        angle = float(input("Enter angle in degrees: "))
        radian = math.radians(angle)
        tangent_value = math.tan(radian)
        r1 = [180 + i * 180 for i in range(100)]
        if angle in r1 or angle==0:
            print("Cotangent is undefined")
        else:
            result = 1 / tangent_value
            print("Cotangent of {} degrees is {:.4f}".format(angle,
↪result))

    elif choice == '7':
        break

    else:
        print("Invalid choice")

    exit_choice = input("Do you wish to continue Trigonometric
↪functions? (y/n): ")
    if exit_choice.lower() == 'n':
        return

def logarithmic_functions():
    while True:
        print("Logarithmic Functions")
        print("1. Natural Logarithm (ln)")

```

```

print("2. Logarithm Base 10")
print("3. Exit to Scientific Calculator")
choice = input("Enter choice: ")

if choice == '1':
    number = float(input("Enter number: "))
    if number <= 0:
        print("Natural logarithm is undefined for non-positive_
↪numbers")
    else:
        result = math.log(number)
        print("Natural logarithm of {} is {:.4f}".format(number,_
↪result))

elif choice == '2':
    number = float(input("Enter number: "))
    if number <= 0:
        print("Logarithm base 10 is undefined for non-positive_
↪numbers")
    else:
        result = math.log10(number)
        print("Logarithm base 10 of {} is {:.4f}".format(number,_
↪result))

elif choice == '3':
    break

else:
    print("Invalid choice")

exit_choice = input("Do you wish to continue to Logarithmic_
↪functions? (y/n): ")
if exit_choice.lower() == 'n':
    return

def other_operations():
    while True:
        print("Other Operations")
        print("1. Factorial")
        print("2. Memory")
        print("3. Exit to Scientific Calculator")
        choice = input("Enter choice: ")

        if choice == '1':
            number = int(input("Enter a non-negative integer: "))
            if number < 0:
                print("Entered number is less than zero")

```

```

        else:
            result = math.factorial(number)
            print("Factorial of {} is {}".format(number, result))

    elif choice == '2':
        global memory
        memory = 0
        print("Memory value:", memory)

    elif choice == '3':
        break
    else:
        print("Invalid choice")

    exit_choice = input("Do you wish to continue to Other operations?_
↪(y/n): ")
    if exit_choice.lower() == 'n':
        return

def math_functions():
    while True:
        print("Mathematical Functions")
        print("1. Ceil")
        print("2. Floor")
        print("3. Modulus")
        print("4. Exit to Scientific Calculator")
        choice = input("Enter choice: ")

        if choice == '1':
            number = float(input("Enter a number: "))
            result = math.ceil(number)
            print("Ceil of {} is {}".format(number, result))

        elif choice == '2':
            number = float(input("Enter a number: "))
            result = math.floor(number)
            print("Floor of {} is {}".format(number, result))

        elif choice == '3':
            dividend = float(input("Enter dividend: "))
            divisor = float(input("Enter divisor: "))
            result = dividend % divisor
            print("Modulus of {} and {} is {}".format(dividend, divisor,
↪result))

        elif choice == '4':
            break

```

```

        else:
            print("Invalid choice")

        exit_choice = input("Do you wish to Mathematical functions? (y/n):_
↪")

        if exit_choice.lower() == 'n':
            return

def scientific_calculator():
    while True:
        print("Welcome to Scientific Calculator")
        print("Select operation:")
        print("1. Trigonometric Functions")
        print("2. Logarithmic Functions")
        print("3. Other Operations")
        print("4. Mathematical Functions")
        print("5. Exit")
        choice = input("Enter choice: ")

        if choice == '1':
            trigonometric_functions()

        elif choice == '2':
            logarithmic_functions()

        elif choice == '3':
            other_operations()

        elif choice == '4':
            math_functions()

        elif choice == '5':
            break

        else:
            print("Invalid choice")

    scientific_calculator()
def conversion():
    def Currency():
        print()
        print("Currency Conversion")
        def usd_to_euro(amount):
            return amount * 0.82

        def usd_to_inr(amount):

```



```
        return amount * 74.67

def usd_to_qar(amount):
    return amount * 3.64

def usd_to_rub(amount):
    return amount * 73.28

def euro_to_usd(amount):
    return amount * 1.22

def euro_to_inr(amount):
    return amount * 90.94

def euro_to_qar(amount):
    return amount * 4.45

def euro_to_rub(amount):
    return amount * 89.96

def inr_to_usd(amount):
    return amount * 0.013

def inr_to_euro(amount):
    return amount * 0.011

def inr_to_qar(amount):
    return amount * 0.049

def inr_to_rub(amount):
    return amount * 0.99

def qar_to_usd(amount):
    return amount * 0.27

def qar_to_euro(amount):
    return amount * 0.22

def qar_to_inr(amount):
    return amount * 20.57

def qar_to_rub(amount):
    return amount * 20.11

def rub_to_usd(amount):
    return amount * 0.014
```

```

def rub_to_euro(amount):
    return amount * 0.011

def rub_to_inr(amount):
    return amount * 1.01

def rub_to_qar(amount):
    return amount * 0.050

def show_menu():
    print("1. Convert from US Dollar")
    print("2. Convert from Euro")
    print("3. Convert from Indian Rupees")
    print("4. Convert from Qatar Riyal")
    print("5. Convert from Russian Ruble")
    print("6. Exit")
def currency()::

    ans="y"
    while ans=="y":
        print()
        show_menu()
        print()
        from_currency = int(input("Enter Choice : "))
        if from_currency>6 or from_currency<=0:
            print("Invalid Choice")
        elif from_currency == 6:
            break
        else:
            print()
            amount = float(input("Enter amount to convert: "))
            if from_currency == 1:
                print("Converted amount to Euro:", usd_to_euro(amount))
                print("Converted amount to Indian Rupees:",
↪usd_to_inr(amount))
                print("Converted amount to Qatar Riyal:",
↪usd_to_qar(amount))
                print("Converted amount Russian Ruble:",
↪usd_to_rub(amount))
            elif from_currency == 2:
                print("Converted amount to USD :", euro_to_usd(amount))
                print("Converted amount to Indian Rupees:",
↪euro_to_inr(amount))
                print("Converted amount to Qatar Riyal:",
↪euro_to_qar(amount))
                print("Converted amount Russian Ruble:",
↪euro_to_rub(amount))

```

```

        elif from_currency == 3:
            print("Converted amount to US Dollar:", □
↳inr_to_usd(amount))
            print("Converted amount to Euro:", inr_to_euro(amount))
            print("Converted amount to Qatar Riyal:", □
↳inr_to_qar(amount))
            print("Converted amount Russian Ruble:", □
↳inr_to_rub(amount))
        elif from_currency == 4:
            print("Converted amount to US Dollar:", □
↳qar_to_usd(amount))
            print("Converted amount to Euro:", qar_to_euro(amount))
            print("Converted amount to Indian Rupees:", □
↳qar_to_inr(amount))
            print("Converted amount Russian Ruble:", □
↳qar_to_rub(amount))
        elif from_currency == 5:
            print("Converted amount to US Dollar:", □
↳rub_to_usd(amount))
            print("Converted amount to Euro:", rub_to_euro(amount))
            print("Converted amount to Indian Rupees:", □
↳rub_to_inr(amount))
            print("Converted amount to Qatar Riyal:", □
↳rub_to_qar(amount))
            print()
            ans=input("Do you want to continue Currency conversions [y/n]□
↳\nPress 'y' to continue : ").lower()
            if __name__ == "__main__":
                currency()

def Length():
    print()
    print("Length Conversion")
    def mm_to_cm(length):
        return length / 10

    def mm_to_m(length):
        return length / 1000

    def mm_to_km(length):
        return length / 1000000

    def cm_to_mm(length):
        return length * 10

    def cm_to_m(length):

```

```

        return length / 100

def cm_to_km(length):
    return length / 100000

def m_to_mm(length):
    return length * 1000

def m_to_cm(length):
    return length * 100

def m_to_km(length):
    return length / 1000

def km_to_mm(length):
    return length * 1000000

def km_to_cm(length):
    return length * 100000

def km_to_m(length):
    return length * 1000

def length():
    ans="y"
    while ans=="y":
        print()
        print("Length Conversion Menu:")
        print("1. Millimeter (mm)")
        print("2. Centimeter (cm)")
        print("3. Meter (m)")
        print("4. Kilometer (km)")
        print("5. Exit")
        print()
        choice = int(input("Enter your choice: "))
        print()
        if choice>5 or choice<=0:
            print("Invalid choice")
        elif choice==5:
            break
        else:
            length = float(input("Enter the length to convert: "))
            print()
            if choice == 1:
                print("Converted lengths:")
                print("Centimeter:", mm_to_cm(length))
                print("Meter:", mm_to_m(length))

```

```

        print("Kilometer:", mm_to_km(length))
    elif choice == 2:
        print("Converted lengths:")
        print("Millimeter:", cm_to_mm(length))
        print("Meter:", cm_to_m(length))
        print("Kilometer:", cm_to_km(length))
    elif choice == 3:
        print("Converted lengths:")
        print("Millimeter:", m_to_mm(length))
        print("Centimeter:", m_to_cm(length))
        print("Kilometer:", m_to_km(length))
    elif choice == 4:
        print("Converted lengths:")
        print("Millimeter:", km_to_mm(length))
        print("Centimeter:", km_to_cm(length))
        print("Meter:", km_to_m(length))
    else:
        print("Invalid choice")
    print()
    ans=input("Do you want to continue length conversions[y/n]_")
    ↪\nPress 'y' to continue : ")

if __name__ == "__main__":
    length()
def Volume():
    print()
    print("Volume Conversion")
    def ml_to_cl(volume):
        return volume / 10

    def ml_to_l(volume):
        return volume / 1000

    def cl_to_ml(volume):
        return volume * 10

    def cl_to_l(volume):
        return volume / 100

    def l_to_ml(volume):
        return volume * 1000

    def l_to_cl(volume):
        return volume * 100

    def volume():
        ans="y"

```

```

while ans=="y":
    print()
    print("Volume Conversion Menu:")
    print("1. Millilitre (ml)")
    print("2. Centilitre (cl)")
    print("3. Litre (l)")
    print("4. Exit")
    print()
    choice = int(input("Enter your choice: "))
    print()
    if choice>4 or choice<=0:
        print("Invalid choice")
    elif choice==4:
        break
    else:
        volume = float(input("Enter the volume to convert: "))
        print()
        if choice == 1:
            print("Converted volumes:")
            print("Centilitre:", ml_to_cl(volume))
            print("Litre:", ml_to_l(volume))
        elif choice == 2:
            print("Converted volumes:")
            print("Millilitre:", cl_to_ml(volume))
            print("Litre:", cl_to_l(volume))
        elif choice == 3:
            print("Converted volumes:")
            print("Millilitre:", l_to_ml(volume))
            print("Centilitre:", l_to_cl(volume))
        else:
            print("Invalid choice")
        print()
        ans=input("Do you want to continue Volume conversions[y/n] ")
↪\nPress 'y' to continue :)
    if __name__ == "__main__":
        volume()
def Mass():
    print()
    print("Mass Conversion")
    def g_to_kg(mass):
        return mass / 1000

    def g_to_mg(mass):
        return mass * 1000

    def g_to_lb(mass):
        return mass / 453.59237

```

```

def kg_to_g(mass):
    return mass * 1000

def kg_to_mg(mass):
    return mass * 1000000

def kg_to_lb(mass):
    return mass * 2.20462

def mg_to_g(mass):
    return mass / 1000

def mg_to_kg(mass):
    return mass / 1000000

def mg_to_lb(mass):
    return mass / 453592.37

def lb_to_g(mass):
    return mass * 453.59237

def lb_to_kg(mass):
    return mass / 2.20462

def lb_to_mg(mass):
    return mass * 453592.37

def mass():
    ans="y"
    while ans=="y":
        print()
        print("Mass Conversion Menu:")
        print("1. Milligram (mg)")
        print("2. Gram (g)")
        print("3. Pound (lbs)")
        print("4. Kilogram (kg)")
        print("5. Exit")
        print()
        choice = int(input("Enter your choice: "))
        print()
        if choice>5 or choice<=0:
            print("Invalid choice")
        elif choice==5:
            break
        else:
            mass = float(input("Enter the mass to convert: "))

```

```

print()
if choice == 2:
    print("Converted masses:")
    print("Kilogram:", g_to_kg(mass))
    print("Milligram:", g_to_mg(mass))
    print("Pound:", g_to_lb(mass))
elif choice == 4:
    print("Converted masses:")
    print("Gram:", kg_to_g(mass))
    print("Milligram:", kg_to_mg(mass))
    print("Pound:", kg_to_lb(mass))
elif choice == 1:
    print("Converted masses:")
    print("Gram:", mg_to_g(mass))
    print("Kilogram:", mg_to_kg(mass))
    print("Pound:", mg_to_lb(mass))
elif choice == 3:
    print("Converted masses:")
    print("Gram:", lb_to_g(mass))
    print("Kilogram:", lb_to_kg(mass))
    print("Milligram:", lb_to_mg(mass))
else:
    print("Invalid choice")
print()
ans=input("Do you want to continue Mass conversions[y/n] ")
↪\nPress 'y' to continue :)
if __name__ == "__main__":
    mass()
def Temp():
    print()
    print("Temperature Conversion")
    def celsius_to_fahrenheit(temp):
        return (temp * 9/5) + 32

    def celsius_to_kelvin(temp):
        return temp + 273.15

    def fahrenheit_to_celsius(temp):
        return (temp - 32) * 5/9

    def fahrenheit_to_kelvin(temp):
        return (temp - 32) * 5/9 + 273.15

    def kelvin_to_celsius(temp):
        return temp - 273.15

    def kelvin_to_fahrenheit(temp):

```



```

        return (temp - 273.15) * 9/5 + 32

def temp():
    ans="y"
    while ans=="y":
        print()
        print("Temperature Conversion Menu:")
        print("1. Celsius (°C)")
        print("2. Fahrenheit (°F)")
        print("3. Kelvin (K)")
        print("4. Exit")
        print()
        choice = int(input("Enter your choice: "))
        print()
        if choice>4 or choice<=0:
            print("Invalid choice")
        elif choice==4:
            break
        else:
            temperature = float(input("Enter the temperature to convert:
↪ "))

            print()
            if choice == 1:
                print("Converted temperatures:")
                print("Fahrenheit:", celsius_to_fahrenheit(temperature))
                print("Kelvin:", celsius_to_kelvin(temperature))
            elif choice == 2:
                print("Converted temperatures:")
                print("Celsius:", fahrenheit_to_celsius(temperature))
                print("Kelvin:", fahrenheit_to_kelvin(temperature))
            elif choice == 3:
                print("Converted temperatures:")
                print("Celsius:", kelvin_to_celsius(temperature))
                print("Fahrenheit:", kelvin_to_fahrenheit(temperature))
            else:
                print("Invalid choice")
            print()
            ans=input("Do you want to continue Temperature conversions[y/n]_
↪ \nPress 'y' to continue :")
    if __name__ == "__main__":
        temp()
def Energy():
    print()
    print("Energy Conversion")
    def joules_to_calories(energy):
        return energy / 4.184

```

```

def joules_to_kilojoules(energy):
    return energy / 1000

def calories_to_joules(energy):
    return energy * 4.184

def calories_to_kilojoules(energy):
    return energy / 239

def kilojoules_to_joules(energy):
    return energy * 1000

def kilojoules_to_calories(energy):
    return energy * 239

def energy():
    ans="y"
    while ans=="y":
        print()
        print("Energy Conversion Menu:")
        print("1. Joules (J)")
        print("2. Kilojoules (kJ)")
        print("3. Calorie (cal)")
        print("4. Exit")
        print()
        choice = int(input("Enter your choice: "))
        print()
        if choice>4 or choice<=0:
            print("Invalid choice")
        elif choice==4:
            break
        else:
            energy = float(input("Enter the energy to convert: "))
            print()
            if choice == 1:
                print("Converted energy:")
                print("Calories:", joules_to_calories(energy))
                print("Kilojoules:", joules_to_kilojoules(energy))
            elif choice == 3:
                print("Converted energy:")
                print("Joules:", calories_to_joules(energy))
                print("Kilojoules:", calories_to_kilojoules(energy))
            elif choice == 2:
                print("Converted energy:")
                print("Joules:", kilojoules_to_joules(energy))
                print("Calories:", kilojoules_to_calories(energy))
            else:

```

```

        print("Invalid choice")
    print()
    ans=input("Do you want to continue Energy conversions[y/n]_")
    ↪\nPress 'y' to continue :")

    if __name__ == "__main__":
        energy()
def Area():
    print()
    print("Area Conversion")
    def sqm_to_sqft(area):
        return area * 10.764

    def sqm_to_sqm(area):
        return area * 1550.0031

    def sqft_to_sqm(area):
        return area / 10.764

    def sqft_to_sqm(area):
        return area * 144

    def sqin_to_sqm(area):
        return area / 1550.0031

    def sqin_to_sqft(area):
        return area / 144

def area():
    ans="y"
    while ans=="y":
        print()
        print("Area Conversion Menu:")
        print("1. Square Inch (sq.in)")
        print("2. Square Feet (sq.ft)")
        print("3. Square Meter (sq.m)")
        print("4. Exit")
        print()
        choice = int(input("Enter your choice: "))
        print()
        if choice>4 or choice<=0:
            print("Invalid choice")
        elif choice==4:
            break
        else:
            area = float(input("Enter the area to convert: "))
            print()

```

```

        if choice == 3:
            print("Converted area:")
            print("Square Feet:", sqm_to_sqft(area))
            print("Square Inch:", sqm_to_sqin(area))
        elif choice == 2:
            print("Converted area:")
            print("Square Meter:", sqft_to_sqm(area))
            print("Square Inch:", sqft_to_sqin(area))
        elif choice == 1:
            print("Converted area:")
            print("Square Meter:", sqin_to_sqm(area))
            print("Square Feet:", sqin_to_sqft(area))
        else:
            print("Invalid choice")

    print()
    ans=input("Do you want to continue Area conversions[y/n]_")
    ↪\nPress 'y' to continue :")

if __name__ == "__main__":
    area()
def main():
    while True:
        print("\nMenu:")
        print("1. Currency")
        print("2. Length")
        print("3. Volume")
        print("4. Mass")
        print("5. Temperature")
        print("6. Energy")
        print("7. Area")
        print("8. Exit")
        print()
        choice = input("Enter your choice: ")
        if choice == '1':
            Currency()
        elif choice == '2':
            Length()
        elif choice == '3':
            Volume()
        elif choice == '4':
            Mass()
        elif choice == '5':
            Temp()
        elif choice == '6':
            Energy()
        elif choice == '7':
            Area()

```

```

        elif choice == '8':
            break
        else:
            print("Invalid choice")

    if __name__ == "__main__":
        main()
#Main_function
answer="y"
while answer=="y":
    print()
    print("VSVC-17180314")
    print("MENU")
    print("1->Standard Operations")
    print("2->Scientific Operations")
    print("3->Programmer Operations")
    print("4->Algebra Operations")
    print("5->Conversion Operations")
    print("6->Exit")
    print()
    choice=int(input("Enter your choice : "))
    if choice==1:
        standard()
    elif choice==2:
        scientific()
    elif choice==3:
        programmer()
    elif choice==4:
        algebra()
    elif choice==5:
        conversion()
    elif choice==6:
        print("Existing!!")
        break
    else:
        print("Invalid Choice")
    print()
    answer=input("Do you want to continue to Main Menu \nPress 'y' to continue :
↵ ")

```

VSVC-17180314

MENU

1->Standard Operations

2->Scientific Operations

3->Programmer Operations

4->Algebra Operations

5->Conversion Operations

6->Exit

Enter your choice : 6

Existing!!

```
[1]: import tkinter as tk

def button_click(value):
    current = entry.get()
    entry.delete(0, tk.END)
    entry.insert(tk.END, current + value)

def clear_entry():
    entry.delete(0, tk.END)

def calculate_result():
    try:
        result = eval(entry.get())
        entry.delete(0, tk.END)
        entry.insert(tk.END, str(result))
    except Exception as e:
        entry.delete(0, tk.END)
        entry.insert(tk.END, "Error")

root = tk.Tk()
root.title("Simple Calculator")

entry = tk.Entry(root, width=20, borderwidth=5)
entry.grid(row=0, column=0, columnspan=4)

buttons = [
    '7', '8', '9', '/',
    '4', '5', '6', '*',
    '1', '2', '3', '-',
    '0', '.', '=', '+',
    '%',
]

row_val = 1
col_val = 0

for button in buttons:
```

```
tk.Button(root, text=button, padx=20, pady=20, command=lambda b=button:
↪button_click(b) if b != '=' else calculate_result() if b == '=' else
↪clear_entry()).grid(row=row_val, column=col_val)
    col_val += 1
    if col_val > 3:
        col_val = 0
        row_val += 1

root.mainloop()
```

[]:

[]: