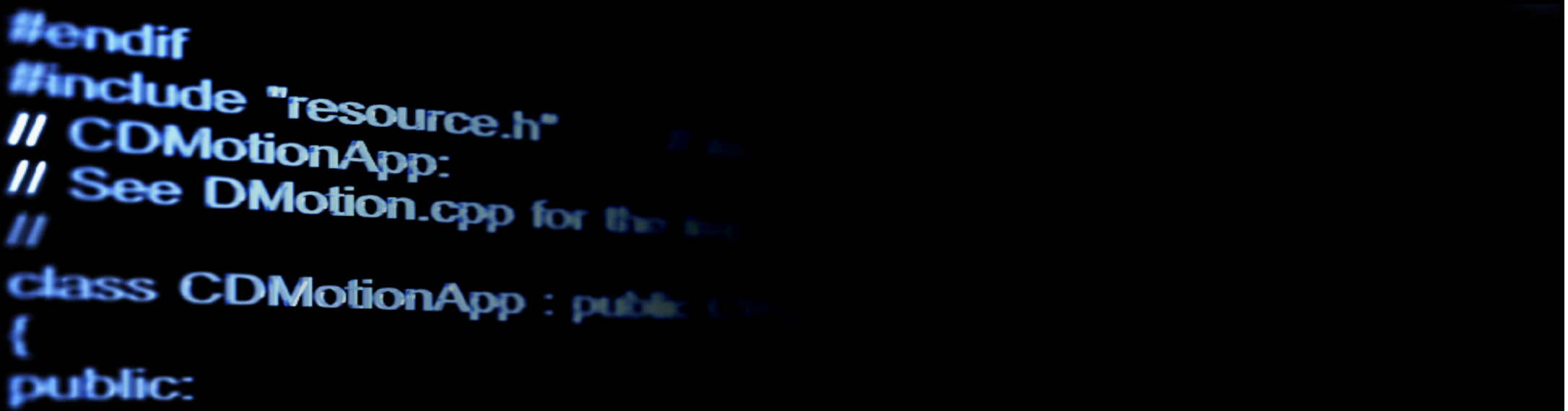
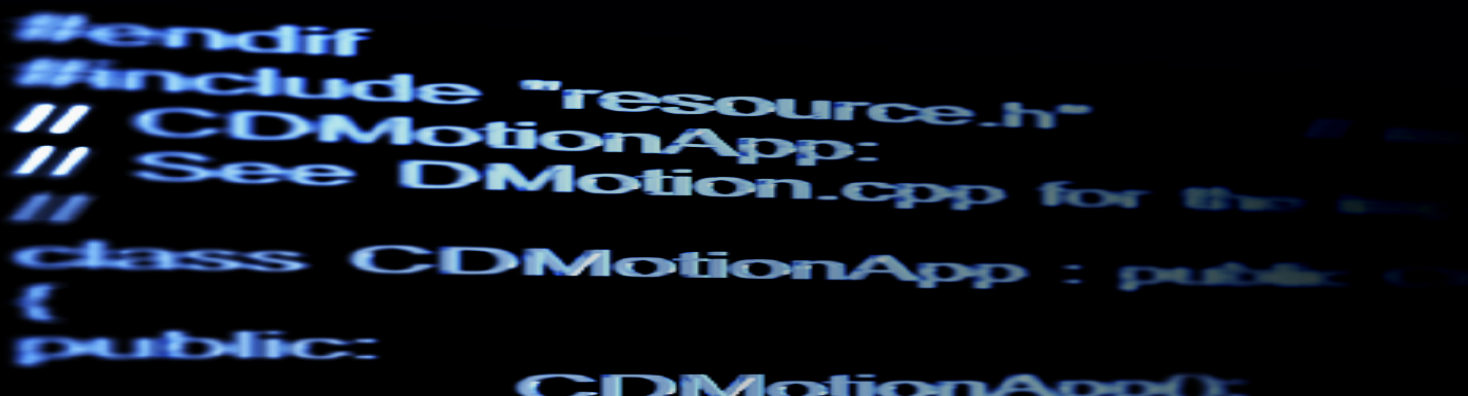
Eigen Value Calculator



Mini Project

```
//Mini Project
on :-->
Eigen Value
Calculator

→Submitted to
'Mrs. Layanvya'

→Submitted by
Arshaan Iqbal
RA2111003011319

(H2)

Medikondur
Koustubh
RA2111003011303

(H2)
        }
```

→Eigenvalues are **the special set of scalar values that is associated with the set of linear equations most probably in the matrix equations**. The eigenvectors are also termed as characteristic roots. It is a non-zero vector that can be changed at most by its scalar factor after the application of linear transformations.



Project Title ⟶ Power Method Using C Programming for Finding Dominant Eigen Value and Eigen Vector

Project Statement ⟶ In this article we are going to develop an **Algorithm for Power Method** for computing largest or dominant Eigen value and corresponding Eigen vector.

Let A be the square matrix of order n i.e. $A_{n \times n}$. Then Power Method starts with one initial approximation to Eigen vector corresponding to largest Eigen value of size n x 1. Let this initial approximation be $X_{n \times 1}$.

After initial assumption, we calculate AX i.e. product of matrix A and X. From the product of AX we divide each element by largest element (by magnitude) and express them as $\lambda_1 X_1$. Obtained value of λ_1 and X_1 are next better approximated value of largest Eigen value and corresponding Eigen vector.

Similarly, for the next step, we multiply A by X_1. From the product of AX_1 we divide each element by largest element (by magnitude) and express them as $\lambda_2 X_2$. Obtained value of λ_2 and X_2 are next better approximated value of largest Eigen value and corresponding Eigen vector.

And then we will repeat this process until largest or dominant Eigen value and corresponding Eigen vector are obtained within desired accuracy

Algorithm for Power Method→

1. Start

2. Read Order of Matrix (n) and Tolerable Error (e)

3. Read Matrix A of Size n x n

4. Read Initial Guess Vector X of Size n x 1

5. Initialize: Lambda_Old = 1

6. Multiply: X_NEW = A * X

7. Replace X by X_NEW

8. Find Largest Element (Lamda_New) by Magnitude from X_NEW

9. Normalize or Divide X by Lamda_New

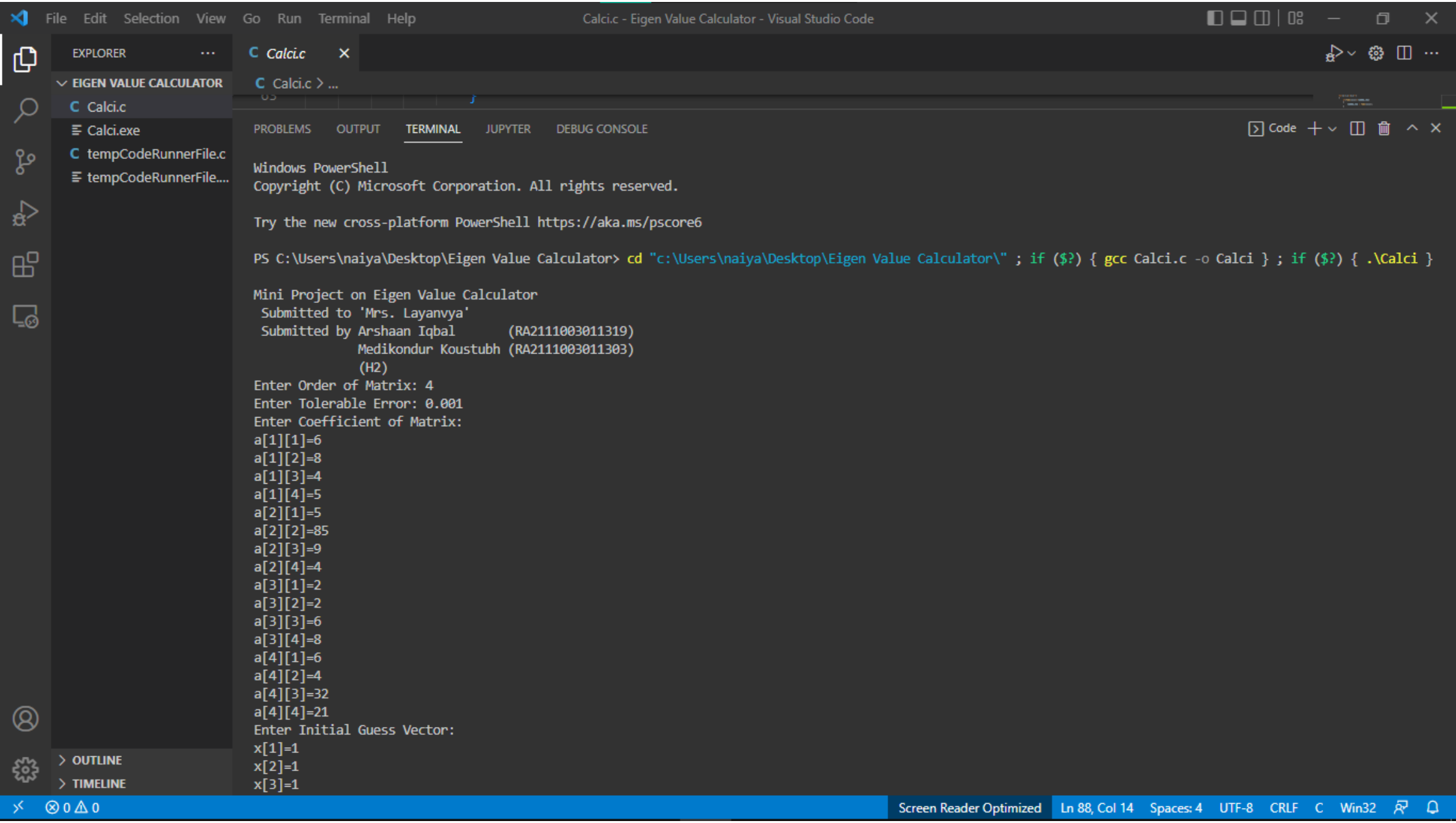
10. Display Lamda_New and X

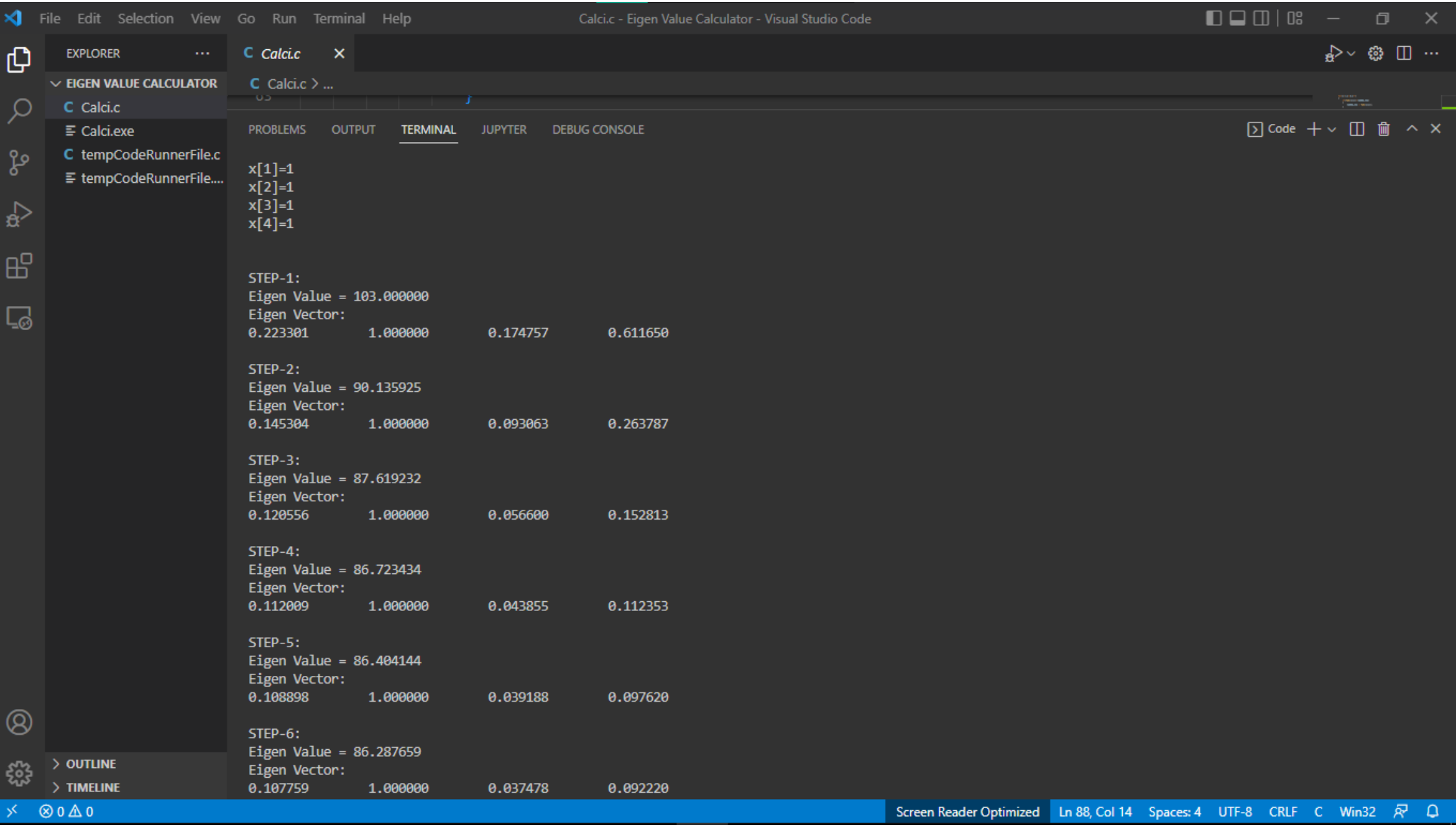
11. If |Lambda_Old - Lamda_New| > e then
set Lambda_Old = Lamda_New and goto
step (6) otherwise goto step (12)

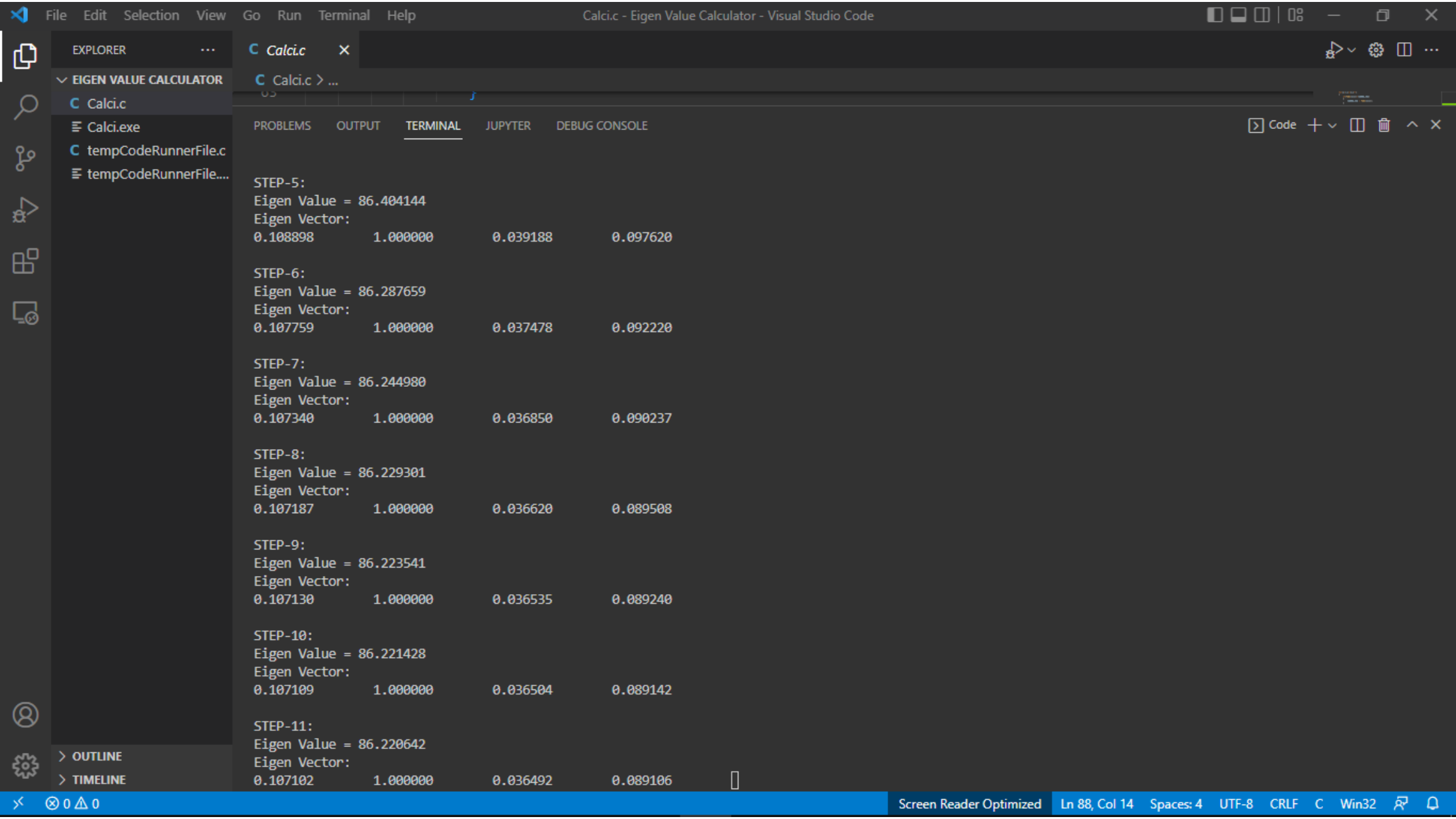
```c
#include <stdio.h>
#include <conio.h>
#include <math.h>

int main()
{
        {
            printf("Mini Project on Eigen Value Calculator\n Submitted to 'Mrs. Layanvya'\n Submitted by Arshaan
Iqbal       (RA2111003011319)\n              Medikondur Koustubh (RA2111003011303)\n              (H2)\n");
        }

            #define SIZE 10
            float a[SIZE][SIZE], x[SIZE],x_new[SIZE];
            float temp, lambda_new, lambda_old, error;
            int i,j,n, step=1;
            //Ask the user to enter Order of Matrix
            printf("Enter Order of Matrix: ");
            scanf("%d", &n);
            //Ask the user to enter Tolerable Error
            printf("Enter Tolerable Error: ");
            scanf("%f", &error);
            //Reading the Matrix
            printf("Enter Coefficient of Matrix:\n");
            for(i=1;i<=n;i++)
            {
                for(j=1;j<=n;j++)
                    {
                        printf("a[%d][%d]=",i,j);
                        scanf("%f", &a[i][j]);
                    }
            }
                // Ask the user to enter Initial guess vector
                printf("Enter Initial Guess Vector:\n");
                for(i=1;i<=n;i++)
                {
```

```
    printf("x[%d]=",i);
    scanf("%f", &x[i]);
}
// Initializing Lambda_Old
lambda_old = 1;
//Multiplying
up:
for(i=1;i<=n;i++)
{
    temp = 0.0;
    for(j=1;j<=n;j++)
    {
        temp = temp + a[i][j]*x[j];
    }
        x_new[i] = temp;
}
//Replacing
for(i=1;i<=n;i++)
{
    x[i] = x_new[i];
}
// Finding largest intg
lambda_new = fabs(x[1]);
for(i=2;i<=n;i++)
{
    if(fabs(x[i])>lambda_new)
    {
        lambda_new = fabs(x[i]);
    }
}
// Normalising
for(i=1;i<=n;i++)
{
    x[i] = x[i]/lambda_new;
}
```

```
            // Final step → Display the output
            printf("\n\nSTEP-%d:\n", step);
            printf("Eigen Value = %f\n", lambda_new);
            printf("Eigen Vector:\n");
            for(i=1;i<=n;i++)
            {
                printf("%f\t", x[i]);
            }
            //Checking Accuracy
            if(fabs(lambda_new-lambda_old)>error)
            {
                lambda_old=lambda_new;
                step++;
            //using goto statement → (up)
                goto up;
            }
            getch();
            return(0);
}
```


File Edit Selection View Go Run Terminal Help
Calci.c - Eigen Value Calculator - Visual Studio Code
EXPLORER
EIGEN VALUE CALCULATOR
Calci.c
Calci.exe
tempCodeRunnerFile.c
tempCodeRunnerFile....
Calci.c
Calci.c > ...
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Code
Windows PowerShell
Copyright (C) Microsoft Corporation. All rights reserved.
Try the new cross-platform PowerShell https://aka.ms/pscore6
PS C:\Users\naiya\Desktop\Eigen Value Calculator> cd "c:\Users\naiya\Desktop\Eigen Value Calculator\" ; if ($?) { gcc Calci.c -o Calci } ; if ($?) { .\Calci }
Mini Project on Eigen Value Calculator
Submitted to 'Mrs. Layanvya'
Submitted by Arshaan Iqbal (RA2111003011319)
Medikondur Koustubh (RA2111003011303)
(H2)
Enter Order of Matrix: 4
Enter Tolerable Error: 0.001
Enter Coefficient of Matrix:
a[1][1]=6
a[1][2]=8
a[1][3]=4
a[1][4]=5
a[2][1]=5
a[2][2]=85
a[2][3]=9
a[2][4]=4
a[3][1]=2
a[3][2]=2
a[3][3]=6
a[3][4]=8
a[4][1]=6
a[4][2]=4
a[4][3]=32
a[4][4]=21
Enter Initial Guess Vector:
x[1]=1
x[2]=1
x[3]=1
OUTLINE
TIMELINE
0 0
Screen Reader Optimized
Ln 88, Col 14
Spaces: 4
UTF-8
CRLF
C
Win32


File Edit Selection View Go Run Terminal Help
Calci.c - Eigen Value Calculator - Visual Studio Code
EXPLORER
EIGEN VALUE CALCULATOR
Calci.c
Calci.exe
tempCodeRunnerFile.c
tempCodeRunnerFile....
Calci.c
Calci.c > ...
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Code
x[1]=1
x[2]=1
x[3]=1
x[4]=1
STEP-1:
Eigen Value = 103.000000
Eigen Vector:
0.223301 1.000000 0.174757 0.611650
STEP-2:
Eigen Value = 90.135925
Eigen Vector:
0.145304 1.000000 0.093063 0.263787
STEP-3:
Eigen Value = 87.619232
Eigen Vector:
0.120556 1.000000 0.056600 0.152813
STEP-4:
Eigen Value = 86.723434
Eigen Vector:
0.112009 1.000000 0.043855 0.112353
STEP-5:
Eigen Value = 86.404144
Eigen Vector:
0.108898 1.000000 0.039188 0.097620
STEP-6:
Eigen Value = 86.287659
Eigen Vector:
0.107759 1.000000 0.037478 0.092220
OUTLINE
TIMELINE
0 0
Screen Reader Optimized
Ln 88, Col 14
Spaces: 4
UTF-8
CRLF
C
Win32


File Edit Selection View Go Run Terminal Help
Calci.c - Eigen Value Calculator - Visual Studio Code
EXPLORER
EIGEN VALUE CALCULATOR
Calci.c
Calci.exe
tempCodeRunnerFile.c
tempCodeRunnerFile....
Calci.c
Calci.c > ...
PROBLEMS OUTPUT TERMINAL JUPYTER DEBUG CONSOLE
Code
STEP-5:
Eigen Value = 86.404144
Eigen Vector:
0.108898 1.000000 0.039188 0.097620
STEP-6:
Eigen Value = 86.287659
Eigen Vector:
0.107759 1.000000 0.037478 0.092220
STEP-7:
Eigen Value = 86.244980
Eigen Vector:
0.107340 1.000000 0.036850 0.090237
STEP-8:
Eigen Value = 86.229301
Eigen Vector:
0.107187 1.000000 0.036620 0.089508
STEP-9:
Eigen Value = 86.223541
Eigen Vector:
0.107130 1.000000 0.036535 0.089240
STEP-10:
Eigen Value = 86.221428
Eigen Vector:
0.107109 1.000000 0.036504 0.089142
STEP-11:
Eigen Value = 86.220642
Eigen Vector:
0.107102 1.000000 0.036492 0.089106
OUTLINE
TIMELINE
0 0
Screen Reader Optimized
Ln 88, Col 14
Spaces: 4
UTF-8
CRLF
C
Win32

Result ⟶ Successfully developed an **Algorithm for Power Method** for computing largest or dominant Eigen value and corresponding Eigen vector.