**Practical : 14**

**Code :**

#include <stdio.h>

#include <math.h>

#include <stdlib.h>

void matrixmult(float matrix[100][100], float vector[], float result[], int n) {

for (int i = 0; i < n; i++) {

result[i] = 0;

for (int j = 0; j < n; j++) {

result[i] += matrix[i][j] \* vector[j];

}

}

}

float vectormagnitude(float vector[], int n) {

float sum = 0;

for (int i = 0; i < n; i++) {

sum += vector[i] \* vector[i];

}

return sqrt(sum);

}

void normalizevector(float vector[], int n) {

float magnitude = vectormagnitude(vector, n);

for (int i = 0; i < n; i++) {

vector[i] /= magnitude;

}

}

float dominanteigenvalue(float matrix[100][100], float initialvector[], float eigenvector[], int n) {

float currentvector[100];

float eigenvalue = 0;

for (int iter = 0; iter < 1000; iter++) {

matrixmult(matrix, initialvector, currentvector, n);

normalizevector(currentvector, n);

float diff = 0;

for (int i = 0; i < n; i++) {

diff += fabs(currentvector[i] - initialvector[i]);

}

if (diff < 1e-6) {

break;

}

for (int i = 0; i < n; i++) {

initialvector[i] = currentvector[i];

}

}

matrixmult(matrix, currentvector, eigenvector, n);

eigenvalue = 0;

for (int i = 0; i < n; i++) {

eigenvalue += eigenvector[i] / currentvector[i];

}

eigenvalue /= n;

for (int i = 0; i < n; i++) {

eigenvector[i] = currentvector[i];

}

return eigenvalue;

}

int main() {

int n;

float matrix[100][100];

float initialvector[100];

float eigenvector[100];

printf("Enter The Order Of The Square Matrix A: ");

scanf("%d", &n);

printf("Enter The Elements Of The Square Matrix A:\n");

for (int i = 0; i < n; i++) {

for (int j = 0; j < n; j++) {

scanf("%f", &matrix[i][j]);

}

}

for (int i = 0; i < n; i++) {

initialvector[i] = 1;

}

float eigenvalue = dominanteigenvalue(matrix, initialvector, eigenvector, n);

printf("Dominant Eigenvalue: %.6f\n", eigenvalue);

printf("Eigen vector:\n");

for (int i = 0; i < n; i++) {

printf("%.6f ", eigenvector[i]);

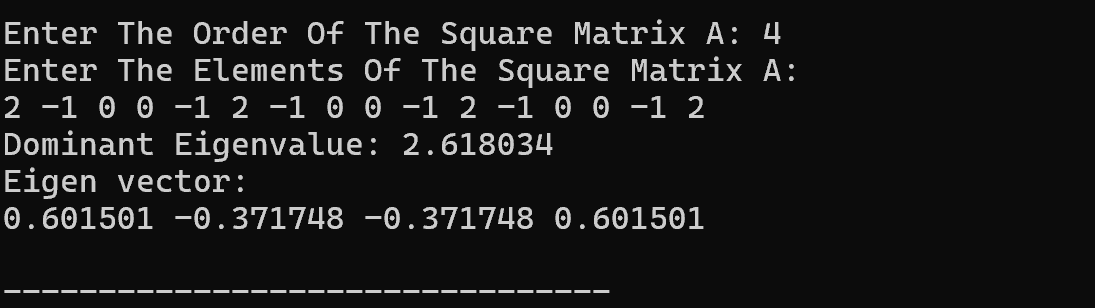
}

printf("\n");

return 0;

}

**Output :**

****