

Varuth Internship Programming Assignment

- 1) Consider a matrix $M_{2 \times 2}$. Write a program that: a) Reads from user: elements of the matrix M b) Check if Eigenvalues and Eigenvectors exists for matrix M c) Find the Eigenvalues and any two Eigenvectors of matrix M d) Prints with suitable message the matrix M , its Eigenvalues and Eigenvectors.

Solution:

```
#include<stdio.h>

#include<math.h>

void main()
{

    int i, j, n=2;

    float A[40][40],x[40],z[40],e[40],zmax,emax;

    /*printf("\nEnter the order of matrix:");

    scanf("%d",&n);*/

    printf("\nEnter matrix elements \n");

    for(i=1; i<=n; i++)
    {
        for(j=1; j<=n; j++)
        {
            printf("A[%d][%d]=", i,j);

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

        }
    }

    printf("\nEnter the column vector\n");

    for(i=1; i<=n; i++)
    {
        printf("X[%d]=",i);

        scanf("%f",&x[i]);
```

```

}
do
{
    for(i=1; i<=n; i++)
    {
        z[i]=0;
        for(j=1; j<=n; j++)
        {
            z[i]=z[i]+A[i][j]*x[j];
        }
    }
    zmax=fabs(z[1]);
    for(i=2; i<=n; i++)
    {
        if((fabs(z[i]))>zmax)
            zmax=fabs(z[i]);
    }
    for(i=1; i<=n; i++)
    {
        z[i]=z[i]/zmax;
    }
    for(i=1; i<=n; i++)
    {
        e[i]=0;
        e[i]=fabs((fabs(z[i]))-(fabs(x[i]))));
    }
    emax=e[1];
    for(i=2; i<=n; i++)
    {

```

```

        if(e[i]>emax)
            emax=e[i];
    }
    for(i=1; i<=n; i++)
    {
        x[i]=z[i];
    }
}
while(emax>0.001);
for(i=1;i<=n;i++)
{
    printf("\n");
    for(j=1;j<=n;j++)
    {
        printf("%f\t",A[i][j]);
    }
    printf("\n");
}
printf("\nThe required eigen value is %f",zmax);
printf("\n\nThe required eigen vector is :\n");
for(i=1; i<=n; i++)
{
    printf("%f\t",z[i]);
}
}

```

Sample Output:

Enter matrix elements

A[1][1]=3

A[1][2]=1

A[2][1]=1

A[2][2]=3

Enter the column vector

X[1]=1

X[2]=0

3.000000 1.000000

1.000000 3.000000

The required eigen value is 3.998049

The required eigen vector is :

1.000000 0.999024

2) A straight line in two dimension can be represented with $y=mx +c$, where y and x are the y and x coordinates of the point on line in two dimension space respectively. m is slope of the line and c is intercept made by the line with y axis. Write a program that: a) Reads from end user using suitable messages : i. Slope m ii. Intercept c iii. Values of x b) Reads from end user using suitable messages: a random point (xp , yp) c) Print with suitable message if (xp, yp) is on the line $y=mx +c$

Solution:

```
#include<stdio.h>
```

```
#include<math.h>
```

```
int main()
```

```
{
```

```
    int m,x,y,c,i,xp,yp;
```

```
    printf("Enter the value of m:\n");                                //m=slope
```

```
    scanf("%d",&m);
```

```
    printf("Enter the value of c:\n");                                //c=intercept
```

```

scanf("%d",&c);
printf("Enter the values of x :\n");
scanf("%d",&x);
y=(m*x)+c;
printf("Enter any random point xp,yp:\n");
scanf("%d %d",&xp,&yp);

if(m==((yp-y)/(xp-x)))                //checking if point is on line
{
    printf("This point is on the line y=mx+c\n");
}
else
{
    printf("Point is not on the line y=mx+c\n");
}
}

```

Sample Output:

Case 1:

Enter the value of m:

5

Enter the value of c:

3

Enter the values of x :

2

Enter any random point xp, yp:

1 8

This point is on the line $y=mx+c$

Case 2:

Enter the value of m:

5

Enter the value of c:

3

Enter the values of x :

2

Enter any random point xp,yp:

1 6

Point is not on the line $y=mx+c$

3) Consider three dimension system. A point in three dimension system is represented as (x, y, z) , where x , y and z are the x , y and z coordinates of the point in three dimension space respectively. Write a program that: a) Reads from end user using suitable messages : i. Point $p1 (x1, y1, z1)$, and ii. Point $p2 (x2, y2, z2)$ b) Prints with suitable message all partial derivatives between the points $(p1, p2)$

Solution:

#Python program to create partial derivatives

```
x1 = int(input("Enter the value of x1 ")) #taking input from user
```

```
y1 = int(input("Enter the value of y1 "))
```

```
z1 = int(input("Enter the value of z1 "))
```

```
x2 = int(input("Enter the value of x2 "))
```

```
y2 = int(input("Enter the value of y2 "))
```

```
z2 = int(input("Enter the value of z2 "))
```

```
p1=(x1,y1,z1)
```

```
p2=(x2,y2,z2)
```

```
print(p1,"and",p2)
```

```
from sympy import Symbol,Derivative
```

```
#printing partial derivatives between p1 and p2
```

```
x=Symbol('x')
```

```
y=Symbol('y')
```

```
z=Symbol('z')
```

```
function=x*x1+y*y1 + z*z1
```

```
partialderiv=Derivative(function,x)
```

```
partialderiv.doit()
```

```
function= x*x1+y*y1 + z*z1
```

```
partialderiv=Derivative(function,y)
```

```
partialderiv.doit()
```

```
function= x*x1+y*y1 + z*z1
```

```
partialderiv=Derivative(function,z)
```

```
partialderiv.doit()
```

```
function=x*x2+y*y2 + z*z2
```

```
partialderiv=Derivative(function,x)
```

```

partialderiv.doit()
function=x*x2+y*y2 + z*z2
partialderiv=Derivative(function,y)
partialderiv.doit()
function==x*x2+y*y2 + z*z2
partialderiv=Derivative(function,z)
partialderiv.doit()

```

Sample output:

```

Enter the value of x1 1
Enter the value of y1 2
Enter the value of z1 3
Enter the value of x2 4
Enter the value of y2 5
Enter the value of z2 6
(1,2,3) and (4,5,6)

```

```

1
2
3
4
5
6

```

5)The area of a square is d. What is the area of the circle which has the diagonal of the square as its diameter? Write a program that: a) Reads from user: area of a square as d (an integer) b) Check if area entered, d is valid

c) Find the area of the circle whose diameter is the same as the diagonal of the square

d) Find the circumference of the circle whose radius is the same as the diagonal of the square

e) Prints with suitable message the area and circumference of the circles respectively

Solution:

```

#include<stdio.h>

#include<math.h>

int main()

{

    int d,area_cir=0,circum=0,n,r;

    printf("Enter the area of square d:\n");

```

```
scanf("%f",&d);

if(d<=0)

{

    printf("Area not valid");

}

else

{

    printf("Valid\n");

}

r=((2*d)/4);

area_cir=(3.142*r);

printf("Area of circle is %f\n",area_cir);


n=sqrt(d);

circum=(2*3.142*(1.414*n));

printf("Circumference of circle is %f\n",circum);

}
```