

SEMESTER – I

MATHEMATICAL PHYSICS PRACTICAL-I

L T P C

18PHU113

- - 4 2

ANY 8 EXPERIMENTS

1. Evaluate the Beta and Gamma function using C program.
2. Write the C Program to Print Square of Each Element of 2D Matrix.
3. Calculate the addition and subtraction of 2 matrices.
4. Write the C Program to Add two Complex Numbers.
5. Write C functions to add and multiply two complex numbers.
6. Program to find Transpose of Given Square Matrix.
7. Evaluate Subtraction of two matrices in C.
8. Write the C Program to perform complex number multiplication.
9. Write the C program to calculate sum of Upper Triangular Elements.
10. Write the C Program to Check whether entered matrix is magic square or not?
11. Find Inverse of a 3 X 3 Matrix using C program.
12. Write C Program to Compute Cross Product of Two Vectors.
13. Write a C Program Friend & Operator: Vector.
14. Write a C Program to Check Prime Number.
15. Check if a Number is Positive or Negative Using if...else.
16. Write a C Program to find the Factorial of a Number.

REFERRECE BOOKS

1. Introduction to Numerical Analysis, S.S. Sastry, 5th Edn., 2012, PHI Learning Pvt. Ltd. Schaum's Outline of Programming with C++. J.Hubbard, 2000, McGraw- Hill Pub.
2. Numerical Recipes in C⁺⁺: The Art of Scientific Computing, W.H. Press et.al, 2nd Edn., 2013, Cambridge University Press.
3. A first course in Numerical Methods, U.M. Ascher & C. Greif, 2012, PHI Learning.
4. An Introduction to computational Physics, T.Pang, 2nd Edn., 2006, Cambridge Univ. Press

List of Experiments

1. Write the C Program to Print Square of Each Element of 2D Matrix.
2. Write the C Program to Add two Complex Numbers.
3. Write the C program to calculate sum of Upper Triangular Elements.
4. Write the C Program to Check whether entered matrix is magic square or not?
5. Find Inverse of a 3 X 3 Matrix using C program.
6. Write C Program to Compute Cross Product of Two Vectors.
7. Write a C Program to Check Prime Number.
8. Check if a Number is Positive or Negative Using if...else.
9. Write a C Program to find the Factorial of a Number.

Aim: Find Inverse of a 3 X 3 Matrix using C program

Program

```
#include<stdio.h>
//          a      3      i      i
void reduction(float a[][6], int size, int pivot, int col)
{
    int i, j;
    float factor;
    factor = a[pivot][col];

    for (i = 0; i < 2 * size; i++) {
        a[pivot][i] /= factor;
    }

    for (i = 0; i < size; i++) {
        if (i != pivot)
        {
            factor = a[i][col];
            for (j = 0; j < 2 * size; j++)
            {
                a[i][j] = a[i][j] - a[pivot][j] * factor;
            }
        }
    }
}

void main()
{
    float matrix[3][6];
    int i, j;

    for (i = 0; i < 3; i++) {
        for (j = 0; j < 6; j++) {
            if (j == i + 3) {
                matrix[i][j] = 1;
            }
        }
    }
}
```

```
else
{
matrix[i][j] = 0;
}
}
}
printf("\nEnter a 3 X 3 Matrix :");
for (i = 0; i < 3; i++) {
for (j = 0; j < 3; j++) {
scanf("%f", &matrix[i][j]);
}
}
for (i = 0; i < 3; i++) {
reduction(matrix, 3, i, i);
}

printf("\nInvers Matrix");
for (i = 0; i < 3; i++) {
printf("\n");
for (j = 0; j < 3; j++) {
printf("%8.3f", matrix[i][j + 3]);
}
}
}
```

Output :

Enter a 3 X 3 Matrix

1 3 1

1 1 2

2 3 4

Invers Matrix

2.000 9.000 -5.000

0.000 -2.000 1.000

-1.000 -3.000 2.000

Aim : To write the C program to calculate sum of Upper Triangular Elements

Program

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

int main() {
    int i, j, a[10][10], sum, rows, columns;

    printf("\nEnter the number of Rows : ");
    scanf("%d", &rows);

    printf("\nEnter the number of Columns : ");
    scanf("%d", &columns);

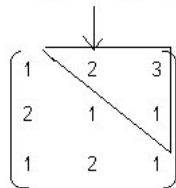
    //Accept the Elements in Matrix
    for (i = 0; i < rows; i++)
        for (j = 0; j < columns; j++) {
            printf("\nEnter the Element a[%d][%d] : ", i, j);
            scanf("%d", &a[i][j]);
        }

    //Addition of all Diagonal Elements
    sum = 0;
    for (i = 0; i < rows; i++)
        for (j = 0; j < columns; j++) {
            // Condition for Upper Triangle
            if (i < j) {
                sum = sum + a[i][j];
            }
        }

    //Print out the Result
    printf("\nSum of Upper Triangle Elements : %d", sum);
    return (0);
}
```

Output

Add these Elements
above main Diagonal



For any element A_{ij}

$$i < j$$

Aim : To write the C Program to Check whether entered matrix is magic square or not

Program

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

int main() {
    int size = 3;
    int matrix[3][3]; // = {{4,9,2},{3,5,7},{8,1,6}};
    int row, column = 0;
    int sum, sum1, sum2;
    int flag = 0;

    printf("\nEnter matrix : ");
    for (row = 0; row < size; row++) {
        for (column = 0; column < size; column++)
            scanf("%d", &matrix[row][column]);
    }

    printf("Entered matrix is : \n");
    for (row = 0; row < size; row++) {
        printf("\n");
        for (column = 0; column < size; column++) {
            printf("\t%d", matrix[row][column]);
        }
    }

    //For diagonal elements
    sum = 0;
    for (row = 0; row < size; row++) {
        for (column = 0; column < size; column++) {
            if (row == column)
                sum = sum + matrix[row][column];
        }
    }

    //For Rows
    for (row = 0; row < size; row++) {
        sum1 = 0;
```

```
    for (column = 0; column < size; column++) {
        sum1 = sum1 + matrix[row][column];
    }
    if (sum == sum1)
        flag = 1;
    else {
        flag = 0;
        break;
    }
}

//For Columns
for (row = 0; row < size; row++) {
    sum2 = 0;
    for (column = 0; column < size; column++) {
        sum2 = sum2 + matrix[column][row];
    }
    if (sum == sum2)
        flag = 1;
    else {
        flag = 0;
        break;
    }
}

if (flag == 1)
    printf("\nMagic square");
else
    printf("\nNo Magic square");

return 0;
}
```

Aim: To write C Program to Compute Cross Product of Two Vectors

Program:

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

const int LOW = 0;
const int HIGH = 10;
int main(int argc, char **argv) {
    time_t seconds;
    time(&seconds);
    srand((unsigned int) seconds);

    int u1, u2, u3, v1, v2, v3;
    u1 = rand() % (HIGH - LOW + 1) + LOW;
    u2 = rand() % (HIGH - LOW + 1) + LOW;
    u3 = rand() % (HIGH - LOW + 1) + LOW;
    v1 = rand() % (HIGH - LOW + 1) + LOW;
    v2 = rand() % (HIGH - LOW + 1) + LOW;
    v3 = rand() % (HIGH - LOW + 1) + LOW;

    int uvi, uvj, uvk;
    uvi = u2 * v3 - v2 * u3;
    uvj = v1 * u3 - u1 * v3;
    uvk = u1 * v2 - v1 * u2;

    printf("The cross product of the 2 vectors \n u = %di + %dj + %dk and \n v = %di + %dj + %dk\n",
        u1, u2, u3, v1, v2, v3);
    printf(" u X v: %di + %dj + %dk", uvi, uvj, uvk);
    return 0;
}
```

Output:

The cross product of the 2 vectors

$u = 0i + 0j + 7k$ and

$v = 4i + 7j + 2k$

$u \times v = -49i + 28j + 0k$

Aim: To write a C Program to find the Factorial of a Number

Program:

```
#include <stdio.h>

int main()
{
    int n, i;
    unsigned long long factorial = 1;
    printf("Enter an integer: ");
    scanf("%d",&n);
    // show error if the user enters a negative integer
    if (n < 0)
        printf("Error! Factorial of a negative number doesn't exist.");
    else
    {
        for(i=1; i<=n; ++i)
        {
            factorial *= i;          // factorial = factorial*i;
        }
        printf("Factorial of %d = %llu", n, factorial);
    }
    return 0;
}
```

Output

Enter an integer: 10

Factorial of 10 = 3628800

Aim : To write a C Program to Check Prime Number

Program:

```
#include <stdio.h>

int main()
{
    int n, i, flag = 0;

    printf("Enter a positive integer: ");
    scanf("%d",&n);

    for(i=2; i<=n/2; ++i)
    {
        // condition for nonprime number
        if(n%i==0)
        {
            flag=1;
            break;
        }
    }

    if (flag==0)
        printf("%d is a prime number.",n);
    else
        printf("%d is not a prime number.",n);

    return 0;
}
```

Output

Enter a positive integer: 29

29 is a prime number.

Aim: To check if a Number is Positive or Negative Using if...else

Program:

```
#include <stdio.h>
int main()
{
    double number;

    printf("Enter a number: ");
    scanf("%lf", &number);

    if (number <= 0.0)
    {
        if (number == 0.0)
            printf("You entered 0.");
        else
            printf("You entered a negative number.");
    }
    else
        printf("You entered a positive number.");
    return 0;
}
```

Output:

Enter a number: 12.3
You entered a positive number.

Aim: To write the C Program to Print Square of Each Element of 2D Matrix

Program:

```
#include<stdio.h>

int main() {
    int M[20][20];
    int m, n, i, j;
    printf("\nEnter the Row and Column sizes of Matrix : ");
    scanf("%d %d", &m, &n);
    printf("\n\n Enter the Matrix elements one by one \n");
    for (i = 1; i<=m; i++)
        for (j = 1; j<=n; j++)
            scanf("%d", &M[i][j]);
    printf("\n The Given Matrix is : \n\n");
    for (i = 1; i<=m; i++){
        for (j = 1; j<=n; j++)
            printf("\t %d", M[i][j]);
        printf("\n\n");
    }
    printf("\n The Square of each elements of Matrix is : \n\n");
    for (i = 1; i<=m; i++){
        for (j = 1; j<=n; j++)
            printf("\t %d", (M[i][j] * M[i][j]));
        printf("\n\n");
    }
    return(0);
}
```

Output:

Enter the Row and Column sizes of Matrix : 3 4

Enter the Matrix elements one by one

0

1

2

3

4

5

6

7

8

9

10

11

The Given Matrix is :

0 1 2 3

4 5 6 7

8 9 10 11

The Square of each elements of Matrix is :

0 1 4 9

16 25 36 49

64 81 100 121

Aim : To write the C Program to Add two Complex Numbers

Program:

```
#include <stdio.h>

struct complex
{
    int real, img;
};

int main()
{
    struct complex a, b, c;
    printf("Enter a and b where a + ib is the first complex number.\n");
    scanf("%d%d", &a.real, &a.img);
    printf("Enter c and d where c + id is the second complex number.\n");
    scanf("%d%d", &b.real, &b.img);
    c.real = a.real + b.real;
    c.img = a.img + b.img;
    printf("Sum of the complex numbers: (%d) + (%di)\n", c.real, c.img);
    return 0;
}
```