

DATA STRUCTURES

MATRIX MULTIPLICATION

1. If A is $m \times n$ matrix such that AB and BA both are defined, then B is a matrix of order
 - a) $n \times n$
 - b) $m \times m$
 - c) $m \times n$
 - d) $n \times m$

2. If A is 3×3 matrix and B is a matrix such that AB and BA are both defined. Then find order of B .
 - (a) 3×2
 - (b) 2×3
 - (c) 2×2
 - (d) 3×3**

3. Which is true about matrix multiplication?
 - (a) It is commutative
 - (b) It is associative**
 - (c) Both (a) and (b)
 - (d) None of these

4. If Matrix X is of order $A \times B$ and Matrix Y is of order $C \times D$, and $B=C$ then the order of the Matrix $X \times Y$ is $A \times D$?
 - a) True**
 - b) False

5. Two matrices A and B are multiplied to get AB if
 - a) both are rectangular
 - b) both have same order
 - c) no. of column of matrix A is equals to no. of Rows of B**
 - d) no of rows of A is equal to no of columns of B

PART-B

1. Write a C program to find product of two given matrices.

To multiply any two matrices in C programming, first ask the user to enter any two matrices, then start multiplying the given two matrices, and store the multiplication result one by one inside any variable, say sum. Store the value of sum in the third matrix (one by one as its element), say mat3, as shown in the program given here.

2. The question is, "Write a program in C that multiplies two given matrices." The answer to this question is given below. This C program asks the user to enter any two 3*3 matrix elements and multiply them to form a new matrix that is the multiplication result of the two given 3*3 matrices. Here, "3*3 matrix" means a matrix that has 3 rows and 3 columns:

```
#include<stdio.h>
#include<conio.h>
int main()
{
    int mat1[3][3], mat2[3][3], mat3[3][3], sum=0, i, j, k;
    printf("Enter first 3*3 matrix element: ");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            scanf("%d", &mat1[i][j]);
    }
    printf("Enter second 3*3 matrix element: ");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            scanf("%d", &mat2[i][j]);
    }
    printf("\nMultiplying two matrices...");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
        {
            sum=0;
            for(k=0; k<3; k++)
                sum = sum + mat1[i][k] * mat2[k][j];
            mat3[i][j] = sum;
        }
    }
    printf("\nMultiplication result of the two given Matrix is: \n");
    for(i=0; i<3; i++)
    {
        for(j=0; j<3; j++)
            printf("%d\t", mat3[i][j]);
        printf("\n");
    }
    getch();
    return 0;
}
```

3. Given an integer matrix with size M x N with odd and even numbers. Separate the even and odd numbers and store in two separate arrays. After separation sort the two 1D arrays in ascending order.
- Sample input: 2 9 12 15 16 24 5 5 7

Sample output: Odd[] = 5 7 9 15 25

Even[] = 2 12 16 24

4. What is the difference between malloc and calloc?

A malloc and calloc are memory management functions. They are used to allocate memory dynamically. Basically, there is no actual difference between calloc and malloc except that the memory that is allocated by calloc is initialized with 0. In C language, calloc function initialize the all allocated space bits with zero but malloc does not initialize the allocated memory. These both function also has a difference regarding their number of arguments, malloc takes one argument but calloc takes two.

5. Is it better to use malloc () or calloc ()?

The calloc function initialize the allocated memory with 0 but malloc don't. So the memory which is allocated by the malloc has the garbage data. In another word you can say that calloc is equal to the combination of malloc and memset.

See the below expression,

```
ptr = calloc(nmember, size); //is essentially equivalent to
```

```
ptr = malloc(nmember * size);
```

```
memset(ptr, 0, (nmember * size));
```

Part-C

Scenario based questions:

1. Write a C program to hold two integer pointers as structure members. Allocate space for two arrays of size 10. Copy the elements of the first array and replace the odd positioned elements by the product of its adjacent elements. Access the array elements and structures using pointers instead of subscript notation.

2. Write a program in C for the following: Get two matrices of varying dimensions. Implement a separate user defined function for multiplying matrices and this function should have recursive call feature to get the results. Print the output matrix in matrix form.