Lab Assignment-2

ROLL: 2005535 | NAME: SAHIL SINGH | DATE: 03/08/21

QUES 1: Write a C Program to merge two sorted array.

SOLUTION:

#include <stdio.h>

#include <stdlib.h>

*int* \*merge(*int* *A*[], *int* *B*[], *int* *size\_array\_1*, *int* *size\_array\_2*)

{

*int* \*array = (*int* \*)calloc(*size\_array\_1* + *size\_array\_2*, sizeof(*int*));

*int* i = 0, j = 0, k = 0;

    while (i < *size\_array\_1* && j < *size\_array\_2*)

    {

        if (*A*[i] < *B*[j])

        {

            array[k++] = *A*[i++];

            continue;

        }

        else if (*A*[i] > *B*[j])

        {

            array[k++] = *B*[j++];

            continue;

        }

        array[k++] = *A*[i++];

        array[k++] = *B*[j++];

    }

    while (i < *size\_array\_1*)

    {

        array[k++] = *A*[i];

        i++;

    }

    while (j < *size\_array\_2*)

    {

        array[k++] = *B*[j];

        j++;

    }

    return array;

}

*int* main()

{

*int* size\_array\_1, size\_array\_2;

    printf("Enter size of the two arrays: ");

    scanf("%d %d", &size\_array\_1, &size\_array\_2);

*int* A[size\_array\_1], B[size\_array\_2];

    printf("Enter in array one (size: %d): ", size\_array\_1);

    for (*int* i = 0; i < size\_array\_1; i++)

    {

        scanf("%d", &A[i]);

    }

    printf("Enter in array two (size: %d): ", size\_array\_2);

    for (*int* j = 0; j < size\_array\_2; j++)

    {

        scanf("%d", &B[j]);

    }

*int* \*array = merge(A, B, size\_array\_1, size\_array\_2);

    for (*int* i = 0; i < size\_array\_1 + size\_array\_2; i++)

    {

        printf("%d ", array[i]);

    }

    return 0;

}

OUTPUT:

Enter the size of two arrays: 5 10

Enter in array 1 of size 5: 1 4 6 7 8

Enter in array 2 of size 10: 1 2 2 3 4 5 8 9 10 14

1 1 2 2 3 4 4 5 6 7 8 8 9 10 14

QUES 2: Write a C Program to find sum and subtraction of two matrices.

SOLUTION:

#include <stdio.h>

*void* readmatrix(*int* *m*[10][10], *int* *row*, *int* *col*)

{

*int* i, j;

    for (i = 0; i < *row*; i++)

    {

        for (j = 0; j < *col*; j++)

        {

            printf("Enter element [%d,%d] : ", i + 1, j + 1);

            scanf("%d", &*m*[i][j]);

        }

    }

}

*void* printmatrix(*int* *m*[10][10], *int* *row*, *int* *col*)

{

*int* i, j;

    for (i = 0; i < *row*; i++)

    {

        for (j = 0; j < *col*; j++)

        {

            printf("%d ", *m*[i][j]);

        }

        printf("\n");

    }

}

*int* main()

{

*int* a[10][10], b[10][10], result[10][10];

*int* i, j, rows1, cols1, rows2, cols2;

    printf("Number of Rows of Matrix 1: ");

    scanf("%d", &rows1);

    printf("Number of Columns of Matrix 1: ");

    scanf("%d", &cols1);

    printf("\nEnter Elements of matrix 1:\n");

    readmatrix(a, rows1, cols1);

    printf("Number of Rows of Matrix 2: ");

    scanf("%d", &rows2);

    printf("Number of Columns of Matrix 2: ");

    scanf("%d", &cols2);

    printf("\nEnter Elements of Matrix 2: \n");

    readmatrix(b, rows2, cols2);

    if (rows1 == rows2 && cols1 == cols2)

    {

        for (i = 0; i < rows1; i++)

        {

            for (j = 0; j < cols1; j++)

            {

                result[i][j] = a[i][j] + b[i][j];

            }

        }

        printf("\nMatrix after adding:\n");

        printmatrix(result, rows1, cols1);

        for (i = 0; i < rows1; i++)

        {

            for (j = 0; j < cols1; j++)

            {

                result[i][j] = a[i][j] - b[i][j];

            }

        }

        printf("\nMatrix after subtracting:\n");

        printmatrix(result, rows1, cols1);

    }

    else

    {

        printf("\nMatrix can not be added, Number of Rows & Columns are Different");

    }

    return 0;

}

OUTPUT:

Number of Rows of Matrix 1: 2

Number of Columns of Matrix 1: 2

Enter Elements of matrix 1:

Enter element [1,1] : 4

Enter element [1,2] : 4

Enter element [2,1] : 3

Enter element [2,2] : 3

Number of Rows of Matrix 2: 2

Number of Columns of Matrix 2: 2

Enter Elements of Matrix 2:

Enter element [1,1] : 2

Enter element [1,2] : 2

Enter element [2,1] : 1

Enter element [2,2] : 1

Matrix after adding:

6 6

4 4

Matrix after subtracting:

2 2

2 2

QUES 3: Write a C program to check two matrices are identical or not.

SOLUTION:

#include <stdio.h>

*int* main()

{

*int* i, j, rows, columns, a[10][10], b[10][10], isEqual;

    printf("\nEnter Number of Rows and Columns: ");

    scanf("%d %d", &i, &j);

    printf("\nEnter the First Matrix Elements:\n");

    for (rows = 0; rows < i; rows++)

    {

        for (columns = 0; columns < j; columns++)

        {

            scanf("%d", &a[rows][columns]);

        }

    }

    printf("\nEnter the Second Matrix Elements\n");

    for (rows = 0; rows < i; rows++)

    {

        for (columns = 0; columns < j; columns++)

        {

            scanf("%d", &b[rows][columns]);

        }

    }

    isEqual = 1;

    for (rows = 0; rows < i; rows++)

    {

        for (columns = 0; columns < j; columns++)

        {

            if (a[rows][columns] != b[rows][columns])

            {

                isEqual = 0;

                break;

            }

        }

    }

    if (isEqual == 1)

    {

        printf("\nMatrix 1 is Equal to Matrix 2");

    }

    else

    {

        printf("\nMatrix 1 is Not Equal to Matrix 2");

    }

    return 0;

}

OUTPUT:

Enter Number of Rows and Columns: 2 2

Enter the First Matrix Elements:

1 2

3 4

Enter the Second Matrix Elements

1 2

3 4

QUES 4: Write a C program to arrange row elements in ascending order.

SOLUTION:

#include <stdio.h>

*void* main()

{

*int* matrix\_a[10][10], matrix\_b[10][10];

*int* i, j, k, a, m, n;

    printf("Enter the number of rows and columns of the matrix: ");

    scanf("%d %d", &m, &n);

    printf("Enter the elements of the matrix: \n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            scanf("%d", &matrix\_a[i][j]);

            matrix\_b[i][j] = matrix\_a[i][j];

        }

    }

    printf("The given matrix is \n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            printf("%d ", matrix\_a[i][j]);

        }

        printf("\n");

    }

    printf("After arranging rows in ascending order\n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            for (k = (j + 1); k < n; ++k)

            {

                if (matrix\_a[i][j] > matrix\_a[i][k])

                {

                    a = matrix\_a[i][j];

                    matrix\_a[i][j] = matrix\_a[i][k];

                    matrix\_a[i][k] = a;

                }

            }

        }

    }

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            printf("%d ", matrix\_a[i][j]);

        }

        printf("\n");

    }

}

OUTPUT:

Enter the number of rows and columns of the matrix: 2 2

Enter the elements of the matrix:

17 7

4 18

The given matrix is

17 7

4 18

After arranging rows in ascending order

7 17

4 18

QUES 5: Write a C program to arrange column elements in ascending order.

SOLUTION:

#include <stdio.h>

*int* main()

{

*int* ma[10][10], mb[10][10];

*int* i, j, k, a, m, n;

    printf("Enter the number of rows and columns of the matrix: \n");

    scanf("%d %d", &m, &n);

    printf("Enter the elements of the matrix: \n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            scanf("%d", &ma[i][j]);

            mb[i][j] = ma[i][j];

        }

    }

    printf("The given matrix is \n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            printf("%d ", ma[i][j]);

        }

        printf("\n");

    }

    printf("After arranging the columns in ascending order \n");

    for (j = 0; j < n; ++j)

    {

        for (i = 0; i < m; ++i)

        {

            for (k = i + 1; k < m; ++k)

            {

                if (ma[i][j] > mb[k][j])

                {

                    a = mb[i][j];

                    mb[i][j] = mb[k][j];

                    mb[k][j] = a;

                }

            }

        }

    }

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            printf("%d ", mb[i][j]);

        }

        printf("\n");

    }

    return 0;

}

OUTPUT:

Enter the number of rows and columns of the matrix:

2 2

Enter the elements of the matrix:

12 15

10 11

The given matrix is

12 15

10 11

After arranging the columns in ascending order

10 11

12 15

QUES 6: Write a C program to find the frequency of even numbers in matrix.

SOLUTION:

#include <stdio.h>

*int* main()

{

*int* array[10][10];

*int* i, j, m, n, even = 0, odd = 0;

    printf("Enter the number of rows and columns of the matrix: \n");

    scanf("%d %d", &m, &n);

    printf("Enter the elements of matrix: \n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            scanf("%d", &array[i][j]);

            if ((array[i][j] % 2) == 0)

            {

                ++even;

            }

            else

                ++odd;

        }

    }

    printf("The given matrix is: \n");

    for (i = 0; i < m; ++i)

    {

        for (j = 0; j < n; ++j)

        {

            printf("%d ", array[i][j]);

        }

        printf("\n");

    }

    printf("\nThe frequency of occurrence of odd number = %d\n", odd);

    printf("The frequency of occurrence of even number = %d\n", even);

    return 0;

}

OUTPUT:

Enter the number of rows and columns of the matrix:

3 3

Enter the elements of matrix:

1 3 1

2 1 1

2 3 1

The given matrix is:

1 3 1

2 1 1

2 3 1

The frequency of occurrence of odd number = 7

The frequency of occurrence of even number = 2

QUES 7: Write a C program to find the trace of (sum of diagonal element) matrix.

SOLUTION:

#include <stdio.h>

*int* main()

{

*int* array[10][10];

*int* i, j, m, n, a = 0, sum = 0;

    printf("Enter the order of the matix \n");

    scanf("%d %d", &m, &n);

    if (m == n)

    {

        printf("Enter the co-efficients of the matrix\n");

        for (i = 0; i < m; ++i)

        {

            for (j = 0; j < n; ++j)

            {

                scanf("%d", &array[i][j]);

            }

        }

        printf("The given matrix is \n");

        for (i = 0; i < m; ++i)

        {

            for (j = 0; j < n; ++j)

            {

                printf("%d ", array[i][j]);

            }

            printf("\n");

        }

        for (i = 0; i < m; ++i)

        {

            sum = sum + array[i][i];

            a = a + array[i][m - i - 1];

        }

        printf("\nThe sum of the main diagonal elements is = %d\n", sum);

        printf("The sum of the off diagonal elements is = %d\n", a);

    }

    else

        printf("The given order is not square matrix\n");

    return 0;

}

OUTPUT:

Enter the order of the matix

2 2

Enter the co-efficients of the matrix

12 14

34 22

The given matrix is

12 14

34 22

The sum of the main diagonal elements is = 34

The sum of the off diagonal elements is = 48

QUES 8: Write a C program to multiply two matrix.

SOLUTION:

#include <stdio.h>

*int* main()

{

*int* m, n, p, q, c, d, k, sum = 0;

*int* first[10][10], second[10][10], multiply[10][10];

    printf("Enter number of rows and columns of first matrix\n");

    scanf("%d%d", &m, &n);

    printf("Enter elements of first matrix\n");

    for (c = 0; c < m; c++)

        for (d = 0; d < n; d++)

            scanf("%d", &first[c][d]);

    printf("Enter number of rows and columns of second matrix\n");

    scanf("%d%d", &p, &q);

    if (n != p)

        printf("The multiplication isn't possible.\n");

    else

    {

        printf("Enter elements of second matrix\n");

        for (c = 0; c < p; c++)

            for (d = 0; d < q; d++)

                scanf("%d", &second[c][d]);

        for (c = 0; c < m; c++)

        {

            for (d = 0; d < q; d++)

            {

                for (k = 0; k < p; k++)

                {

                    sum = sum + first[c][k] \* second[k][d];

                }

                multiply[c][d] = sum;

                sum = 0;

            }

        }

        printf("Product of the matrices:\n");

        for (c = 0; c < m; c++)

        {

            for (d = 0; d < q; d++)

                printf("%d ", multiply[c][d]);

            printf("\n");

        }

    }

    return 0;

}

OUTPUT:

Enter number of rows and columns of first matrix

2 2

Enter elements of first matrix

1 2

2 1

Enter number of rows and columns of second matrix

2 2

Enter elements of second matrix

1 2

2 1

Product of the matrices:

5 4

4 5

QUES 9: Write a C program to create a structure of Employee having data members ID, Name, Age, Salary. Create array of structure of employee of size n by dynamic memory allocation. Assign pointer to the structure and find the highest salary of the employee and avgSalary().

SOLUTION:

#include <stdio.h>

#include <stdlib.h>

*struct* employee

{

*int* id, age;

*char* name[100];

*float* salary;

};

*float* avgSalary(*struct* employee \**a*, *int* *n*)

{

*float* avgSalary = 0;

    for (*int* i = 0; i < *n*; i++)

    {

        avgSalary += *a*[i].salary;

    }

    avgSalary /= (*float*)*n*;

    return avgSalary;

}

*float* highSalary(*struct* employee \**a*, *int* *n*)

{

*float* high = -2147483648;

    for (*int* i = 0; i < *n*; i++)

    {

        if (high < *a*[i].salary)

        {

            high = *a*[i].salary;

        }

    }

    return high;

}

*int* main()

{

*int* n;

    printf("Enter number of employees: ");

    scanf("%d", &n);

*struct* employee \*a;

    a = (*struct* employee \*)malloc(n \* sizeof(*struct* employee));

    for (*int* i = 0; i < n; i++)

    {

        printf("\nEnter employee id: ");

        scanf("%d", &a[i].id);

        printf("Enter employee name: ");

        getchar();

        fgets(a[i].name, 100, stdin);

        printf("Enter employee age: ");

        scanf(" %d", &a[i].age);

        printf("Enter employee salary: ");

        scanf("%f", &a[i].salary);

    }

    printf("Average Salary: %0.2f\n", avgSalary(a, n));

    printf("Highest Salary: %0.2f\n", highSalary(a, n));

    return 0;

}

OUTPUT:

Enter number of employees: 2

Enter employee id: 2005535

Enter employee name: SAHIL SINGH

Enter employee age: 19

Enter employee salary: 200000

Enter employee id: 1080394

Enter employee name: KIIT

Enter employee age: 26

Enter employee salary: 2000000

Average Salary: 1100000.00

Highest Salary: 2000000.00