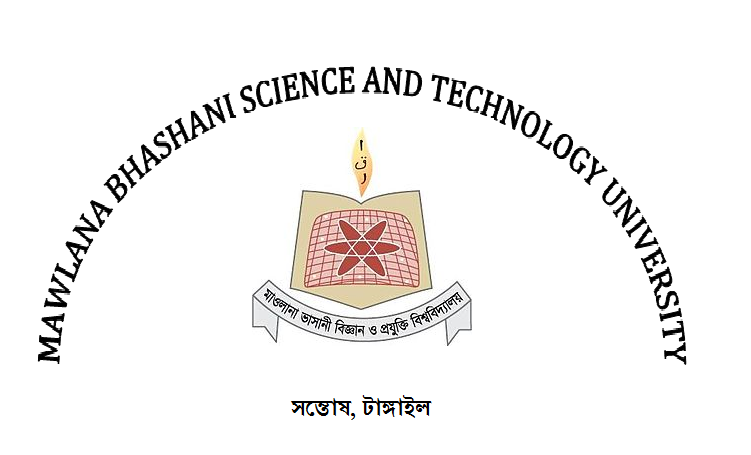
**Lab Report-01**

**Report Name : Different matrix program**  
**Course Code : ICT-2108**  
**Course Title : Algorithm Design and Analysis Lab Course Credit : 1.00**

|  |  |
| --- | --- |
| Submitted by, **Name :MOHAMMAD MEHEDY HASAN RIDOY**  **ID : IT-16024**  **Batch :2ND YEAR 1st  SEMESTER**  **Dept. Of INFORMATION AND COMMUNICATION TECHNOLOGY** | Submitted to, **MOHAMMAD BADRUL ALAM MIAH**  **ASSISTANT PROFESSOR**  **Dept. Of INFORMATION AND COMMUNICATION TECHNOLOGY**  **MAWLANA BHASHANI SCIENCE AND TECHNOLOGY UNIVERSITY** |

**Date of performance: 20.02.2017 Date of Submission: 26.02.2017**

Problem no: 01

**Write a program of sum of two matrix**

Pseudo code:

#include <bits/stdc++.h>

using namespace std;

int main()

{

int m, n, c, d, first[100][100], second[100][100], sum[100][100];

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

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

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

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

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

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

for (c = 0; c < m; c++)

for (d = 0 ; d < n; d++)

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

printf("Sum of entered matrices:-\n");

for (c = 0; c < m; c++)

{

for (d = 0 ; d < n; d++)

{

sum[c][d] = first[c][d] + second[c][d];

printf("%d\t", sum[c][d]);

}

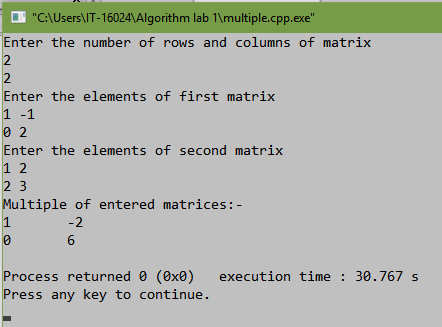
printf("\n");

}

return 0;

}

Output:



Program no:02

**Write a program of subtract of matrix.**

Pseudo code:

#include <bits/stdc++.h>

using namespace std;

int main()

{

int m, n, c, d, first[100][100], second[100][100], sum[100][100];

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

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

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

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

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

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

for (c = 0; c < m; c++)

for (d = 0 ; d < n; d++)

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

printf("Subtraction of entered matrices:-\n");

for (c = 0; c < m; c++)

{

for (d = 0 ; d < n; d++)

{

sum[c][d] = first[c][d] - second[c][d];

printf("%d\t", sum[c][d]);

}

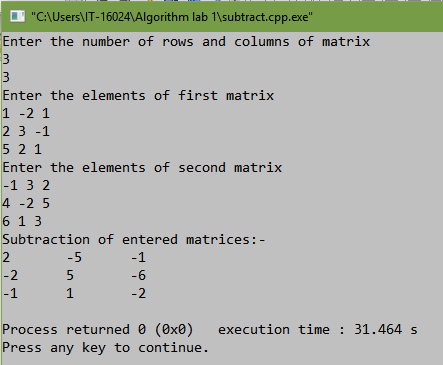
printf("\n");

}

return 0;

}

Output:



Problem no: 03

**Write a program to multiply two matrix**.

Pseudo code:

#include <bits/stdc++.h>

using namespace std;

int main()

{

int m, n, c, d, first[100][100], second[100][100], sum[100][100];

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

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

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

for (c = 0; c < m; c++)

for (d = 0; d < n; d++)

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

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

for (c = 0; c < m; c++)

for (d = 0 ; d < n; d++)

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

printf("Multiple of entered matrices:-\n");

for (c = 0; c < m; c++)

{

for (d = 0 ; d < n; d++)

{

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

printf("%d\t", sum[c][d]);

}

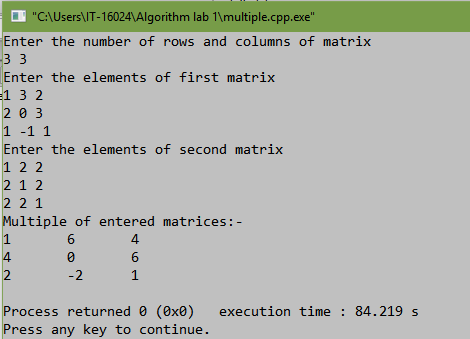
printf("\n");

}

return 0;

}

Output:



Problem no: 04

**Write a program of square matrix (both for diagonal & boundary)**

Pseudo code:

#include<stdio.h>

int main()

{

int a[100][100],i,j,sum=0,m,n;

printf("\nEnter the row and column of matrix: ");

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

printf("\nEnter the elements of matrix: ");

for(i=0; i<m; i++)

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

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

printf("\nThe matrix is\n");

for(i=0; i<m; i++)

{

printf("\n");

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

{

printf("%d\t",a[i][j]);

}

}

for(i=0; i<m; i++)

{

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

{

if(i==j || (i+j==n-1) || (i==0 || j==0) || (i==n-1 || j==n-1))

sum=sum+a[i][j];

}

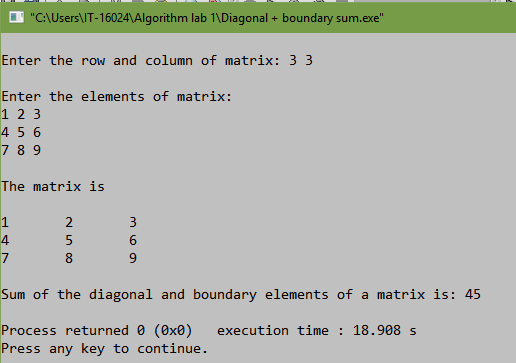
}

printf("\n\nSum of the diagonal and boundary elements of a matrix is: %d\n",sum);

return 0;

}

Output:



Problem no: 05

Write a program of transpose of a matrix.

Pseudo code:

#include <stdio.h>

int main()

{

int a[10][10], transpose[10][10], r, c, i, j;

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

scanf("%d %d", &r, &c);

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

for(i=0; i<r; ++i)

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

{

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

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

}

printf("\nEntered Matrix: \n");

for(i=0; i<r; ++i)

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

{

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

if (j == c-1)

printf("\n\n");

}

for(i=0; i<r; ++i)

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

{

transpose[j][i] = a[i][j];

}

printf("\nTranspose of Matrix:\n");

for(i=0; i<c; ++i)

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

{

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

if(j==r-1)

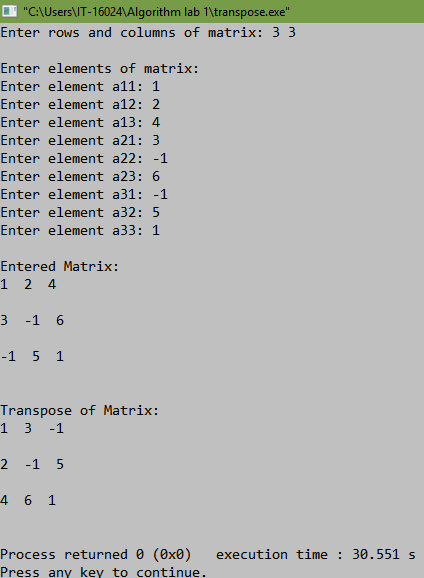
printf("\n\n");

}

return 0;

}

Output:



Problem no:06

**Write a program of inverse of a matrix.**

Pseudo code:

#include<stdio.h>

#include<math.h>

float determinant(float [][25], float);

void cofactor(float [][25], float);

void transpose(float [][25], float [][25], float);

int main()

{

float a[25][25], k, d;

int i, j;

printf("Enter the order of the Matrix : ");

scanf("%f", &k);

printf("Enter the elements of %.0fX%.0f Matrix : \n", k, k);

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

{

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

{

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

}

}

d = determinant(a, k);

if (d == 0)

printf("\nInverse of Entered Matrix is not possible\n");

else

cofactor(a, k);

}

float determinant(float a[25][25], float k)

{

float s = 1, det = 0, b[25][25];

int i, j, m, n, c;

if (k == 1)

{

return (a[0][0]);

}

else

{

det = 0;

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

{

m = 0;

n = 0;

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

{

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

{

b[i][j] = 0;

if (i != 0 && j != c)

{

b[m][n] = a[i][j];

if (n < (k - 2))

n++;

else

{

n = 0;

m++;

}

}

}

}

det = det + s \* (a[0][c] \* determinant(b, k - 1));

s = -1 \* s;

}

}

return (det);

}

void cofactor(float num[25][25], float f)

{

float b[25][25], fac[25][25];

int p, q, m, n, i, j;

for (q = 0; q < f; q++)

{

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

{

m = 0;

n = 0;

for (i = 0; i < f; i++)

{

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

{

if (i != q && j != p)

{

b[m][n] = num[i][j];

if (n < (f - 2))

n++;

else

{

n = 0;

m++;

}

}

}

}

fac[q][p] = pow(-1, q + p) \* determinant(b, f - 1);

}

}

transpose(num, fac, f);

}

void transpose(float num[25][25], float fac[25][25], float r)

{

int i, j;

float b[25][25], inverse[25][25], d;

for (i = 0; i < r; i++)

{

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

{

b[i][j] = fac[j][i];

}

}

d = determinant(num, r);

for (i = 0; i < r; i++)

{

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

{

inverse[i][j] = b[i][j] / d;

}

}

printf("\n\n\nThe inverse of matrix is : \n");

for (i = 0; i < r; i++)

{

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

{

printf("\t%f", inverse[i][j]);

}

printf("\n");

}

}

Output:

