FUNDAMENTALS OF PROGRAMMING(LAB)

By

1.MUHAMMAD AHMED (455401-ME-15)

Lab Instructor: MUHAMMAD AFFAN
Subject: FUNDAMENTALS OF PROGRAMMING
Date: DEC-14-2023

_____

**SCHOOL OF MECHANICAL
AND MANUFACTURING ENGINEERING,
NATIONAL UNIVERSITY OF
SCIENCES AND TECHNOLOGY
ISLAMABAD, PAKISTAN.**

LAB TASK

Q1.

```cpp
#include <iostream>

using namespace std;

int main()

{

int mat[3][3];

int sum = 0;

for (int i = 0; i < 3;i++)

{

cout << "Enter data for row " << i + 1 << endl;

for (int j = 0; j < 3;j++)

{

cin >> mat[i][j];

}

}

cout << "The resultant matrix\n";

for (int m = 0; m < 3; m++)

{

for (int n = 0;n < 3; n++)

{

cout << mat[m][n]<<" ";

}

cout << endl;

}

for (int k = 0; k < 3;k++)

{

sum += mat[k][k];

}

cout << "The sum of the right diagonal is " << sum << endl;

sum = 0;
```

```cpp
for (int z = 2;z>=0 ;z--)

{

sum += mat[2-z][z];

}

cout << "The sum of the left diagonal is " << sum;

return 0;}
```

Q2.

```cpp
#include <iostream>

using namespace std;

int result(int a[3][3], int b[3][3])

{

int sum[3][3];

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

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

}

}

cout << "The resultant matrix after addition is: " << endl;

for (int i = 0; i < 3; i++)

{

for (int j = 0; j < 3; j++)

{

cout << sum[i][j]<<" ";

}

cout << endl;

}
```

```cpp
    return 0;

}

int main()

{

int num = 0;

int mat1[3][3];

int mat2[3][3];

cout << "Enter the data for matrix number one\n";

for (int i = 0; i < 3;i++)

{

cout << "Enter data for row " << i + 1 << endl;

for (int j = 0; j < 3;j++)

{

cin >> mat1[i][j];

}

}

cout << "The resultant matrix\n";

for (int m = 0; m < 3; m++)

{

for (int n = 0;n < 3; n++)

{

cout << mat1[m][n]<<" ";

}

cout << endl;

}

cout << "Enter the data for matrix number two\n";

for (int i = 0; i < 3;i++)

{

cout << "Enter data for row " << i + 1 << endl;

for (int j = 0; j < 3;j++)

{
```

```cpp
cin >> mat2[i][j];

}

}

cout << "The resultant matrix\n";

for (int m = 0; m < 3; m++)

{

for (int n = 0;n < 3; n++)

{

cout << mat2[m][n] << " ";

}

cout << endl;

}

result(mat1, mat2);

return 0;


}
```

Q3.

```cpp
#include <iostream>

using namespace std;

int transpose(int a[3][3])

{

int result[3][3];

for (int i = 0; i < 3;i++)

{

for (int j = 0; j < 3; j++)

{

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

}

}
```

```cpp
cout << "While the transpose of the matrix is: " << endl;

for (int i = 0; i < 3;i++)

{

for (int j = 0; j < 3; j++)

{

cout << result[i][j]<<" ";

}

cout << "\n";

}

return 0;

}

int main()

{

int mat[3][3];

for (int i = 0; i < 3;i++)

{

cout << "Enter data of row " << i + 1 << endl;

for (int j = 0; j < 3; j++)

{

cin >> mat[i][j];

}

}

cout << "The resultant matrix is " << endl;

for (int i = 0; i < 3;i++)

{

for (int j = 0; j < 3; j++)

{

cout<< mat[i][j]<<" ";

}

cout << endl;

}
```

```cpp
cout << endl;

transpose(mat);

return 0;

}
```

Q4.

```cpp
#include <iostream>

using namespace std;

int multiply(int a[2][2], int b[2][2])

{

int result[2][2] = { 0 };

for (int i = 0; i < 2;i++)

{

for (int j = 0; j < 2; j++)

{

for (int k = 0; k < 2;k++)

{

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

}

}

}

cout << "The result of multiplication is: " << endl;

for (int i = 0; i < 2;i++)

{

for (int j = 0; j < 2; j++)

{

cout << result[i][j]<<" ";

}

cout << endl;

}

return 0;
```

```cpp
}
int main()
{
int mat1[2][2];
cout << "Enter data for matrix 1\n";
for (int i = 0; i < 2;i++)
{
cout << "Enter data of row " << i + 1 << endl;
for (int j = 0; j < 2; j++)
{
cin >> mat1[i][j];
}
}
cout << "The resultant matrix is " << endl;
for (int i = 0; i < 2;i++)
{
for (int j = 0; j < 2; j++)
{
cout<< mat1[i][j]<<" ";
}
cout << endl;
}
cout << endl;
int mat2[2][2];
cout << "Enter data for matrix 2\n";
for (int i = 0; i < 2;i++)
{
cout << "Enter data of row " << i + 1 << endl;
for (int j = 0; j < 2; j++)
{
cin >> mat2[i][j];
```

```cpp
}
}
cout << "The resultant matrix is " << endl;
for (int i = 0; i < 2;i++)
{
for (int j = 0; j < 2; j++)
{
cout << mat2[i][j] << " ";
}
cout << endl;
}
cout << endl;
multiply(mat1, mat2);
return 0;
}
```

Q5.

```cpp
#include <iostream>
using namespace std;
int table(int num, int i)
{
 if (i <= 10)
 {
 cout << num << "*" << i << "=" << num * i << endl;
 return table(num, i + 1);
 }
 else
 {
 return 1;
 }
}
int main()
```

```cpp
{
 table(15, 1);
 return 0;}
```

HOME TASK

Q1.

```cpp
#include <iostream>
using namespace std;
int Determinant(int a[2][2])
{
 return (a[0][0] * a[1][1]) - (a[0][1] * a[1][0]);
}
int Adjoint(int b[2][2])
{
 int temp = 0;
 temp = b[0][0];
 b[0][0] = b[1][1];
 b[1][1] = temp;
 temp = -b[0][1];
 b[0][1] = -b[1][0];
 b[1][0] = temp;
 return 0;
}
int Inverse(int a[2][2])
{
 cout << "The inverse is :" << endl;
 cout << 1 << "/" << Determinant(a) << " multiplied by the matrix :" << endl;
 for (int i = 0; i < 2;i++)
 {
 for (int j = 0; j < 2; j++)
 {
```

```cpp
cout << a[i][j] << " ";
}
cout << endl;
}
return 0;
}
int InputMatrix(int b[2][2])
{
for (int i = 0; i < 2; i++)
{
cout << "Enter data for row " << i + 1 << endl;
for (int j = 0; j < 2; j++)
{
cin >> b[i][j];
}
}
return 0;
}
int main()
{
int mat[2][2];
InputMatrix(mat);
cout << "The input matrix :" << endl;
for (int i = 0; i < 2;i++)
{
for (int j = 0; j < 2; j++)
{
cout << mat[i][j] << " ";
}
cout << endl;
}
```

```
Determinant(mat);

Adjoint(mat);

Inverse(mat);

return 0;

}}
```