DSA Lab02

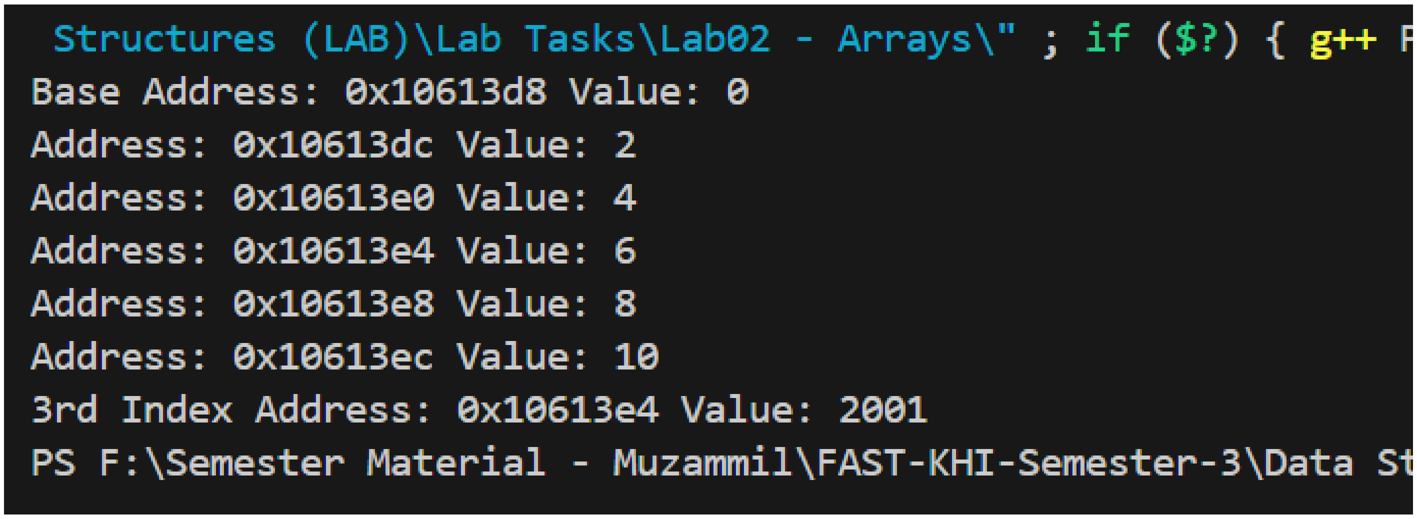
23K2001

M.Muzammil Siddiqui

BCS-3J

Practice Task:

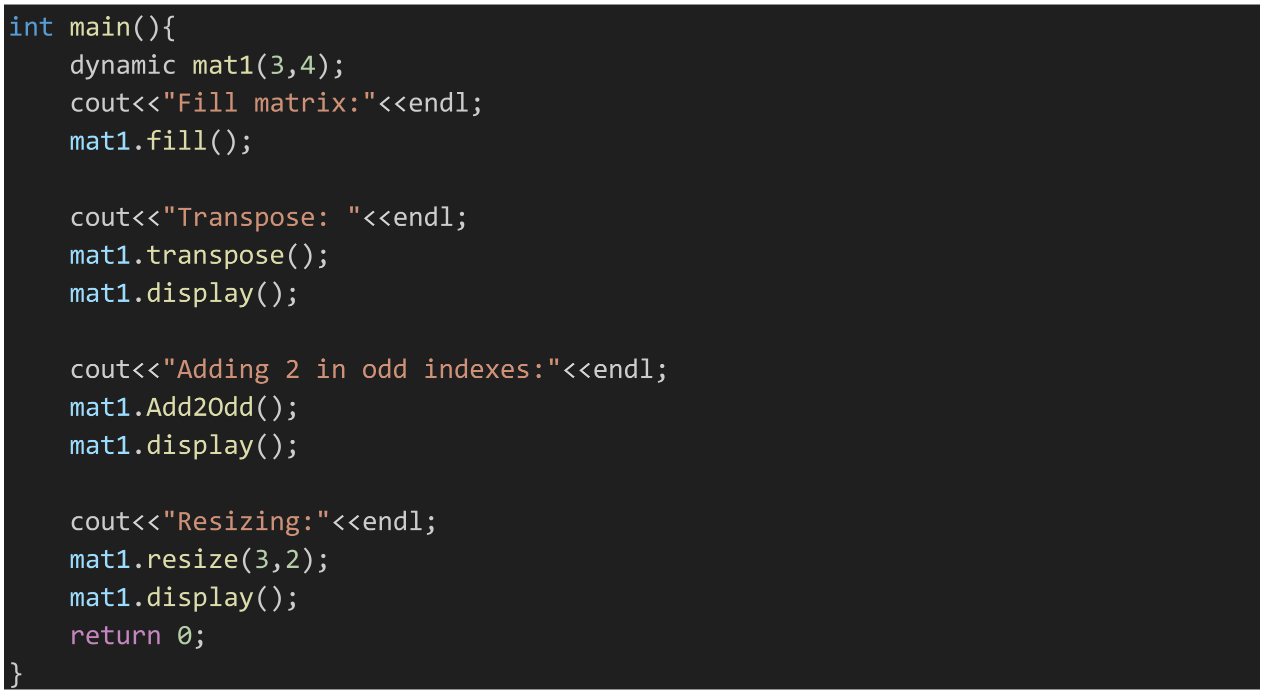
```cpp
//23K2001 - Muzammil
#include<iostream>
using namespace std;
int main(){
    int *num;
    num = new int[6];
    for(int i=0,j=0;i<6;i++,j+=2)
    num[i]=j;

    cout<<"Base Address: "<<num<<" Value: "<<*num<<endl;
    for(int i=1;i<6;i++)
    cout<<"Address: "<<num+i<<" Value: "<<num[i]<<endl;

    num[3] = 2001;
    cout<<"3rd Index Address: "<<num+3<<" Value:
"<<num[3]<<endl;

    delete[] num;

    return 0;
}
```



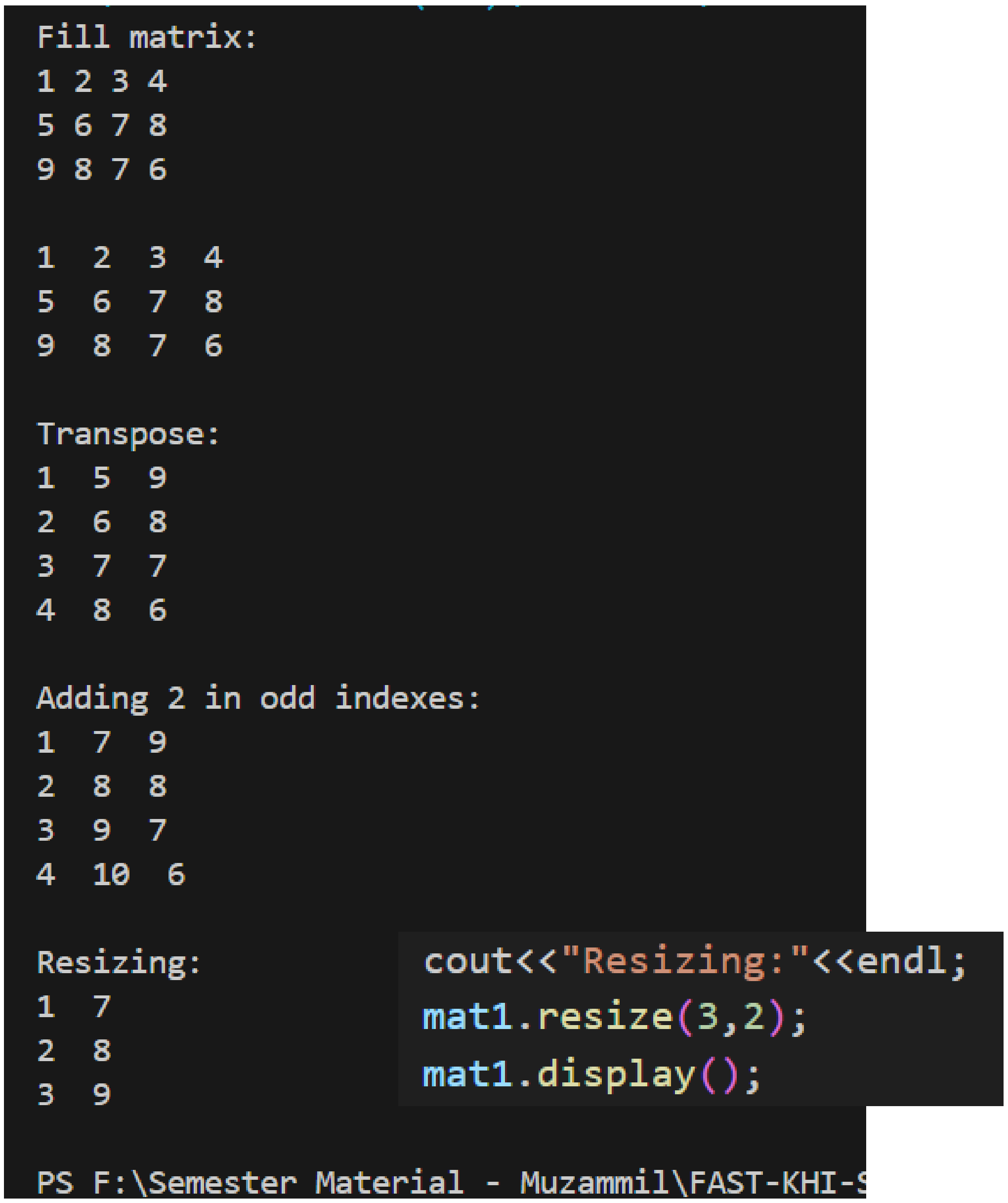
```
 Structures (LAB)\Lab Tasks\Lab02 - Arrays\" ; if ($?) { g++ F
Base Address: 0x10613d8 Value: 0
Address: 0x10613dc Value: 2
Address: 0x10613e0 Value: 4
Address: 0x10613e4 Value: 6
Address: 0x10613e8 Value: 8
Address: 0x10613ec Value: 10
3rd Index Address: 0x10613e4 Value: 2001
PS F:\Semester Material - Muzammil\FAST-KHI-Semester-3\Data St
```

Q1:

```cpp
//23K2001 - Muzammil
#include<iostream>
using namespace std;

class dynamic{
    private:
    int rows,cols;
    int **matrix;

    public:
    dynamic(){matrix=NULL;}
    dynamic(int r,int c,int val=0){
        rows = r;
        cols = c;
        matrix = new int*[rows];

        for(int i=0;i<rows;i++){
        matrix[i] = new int[cols];
        for(int j=0;j<cols;j++)
            matrix[i][j] = val;
        }
    }

    void resize(int r,int c,int val=0){
        int **old = new int*[r];
        for(int i=0;i<r;i++){
        old[i] = new int[c];
        for(int j=0;j<c;j++)
            old[i][j] = val;
        }

        for(int i=0;i<(rows < r ? rows: r);i++)
            for(int j=0;j<(cols < c ? cols: c);j++)
                old[i][j] = matrix[i][j];

        for(int i=0;i<rows;i++)
        delete[] matrix[i];

        delete[] matrix;

        if(rows<r || cols<c){
            matrix = new int*[r];
            for(int i=0;i<r;i++)
            matrix[i] = new int[c];

            for(int i=0;i<r;i++)
                for(int j=0;j<c;j++)
                    matrix[i][j] = old[i][j];
        }
        else{
            matrix = new int*[r];
            for(int i=0;i<r;i++){
                matrix[i] = new int[c];
                for(int j=0;j<c;j++)
                matrix[i][j] = old[i][j];
            }
        }

        for(int i=0;i<rows;i++)
            delete[] old[i];
        delete[] old;

        rows=r;
        cols=c;
    }

    void transpose(){
        int **old = new int*[rows];
        for(int i=0;i<rows;i++){
            old[i] = new int[cols];
            for(int j=0;j<cols;j++)
            old[i][j] = matrix[i][j];
        }

        for(int i=0;i<rows;i++)
            delete[] matrix[i];

        delete[] matrix;

        matrix = new int*[cols];
        for(int i=0;i<cols;i++){
            matrix[i] = new int[rows];
            for(int j=0;j<rows;j++)
            matrix[i][j] = old[j][i];
        }
        for(int i=0;i<rows;i++)
            delete[] old[i];

        delete[] old;

        rows += cols;
        cols = rows-cols;
        rows -= cols;
    }

    void fill(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<cols;j++)
            cin>>matrix[i][j];
        }
        cout<<endl;
        display();
    }

    void display(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<cols;j++)
            cout<<matrix[i][j]<<"   ";

            cout<<endl;
        }
        cout<<endl;
    }

    void Add2Odd(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<cols;j++){
                if(j%2!=0)
                    matrix[i][j]+=2;
            }
        }
    }

    ~dynamic(){
        for(int i=0;i<rows;i++)
            delete[] matrix[i];
        delete[] matrix;
    }
};
```

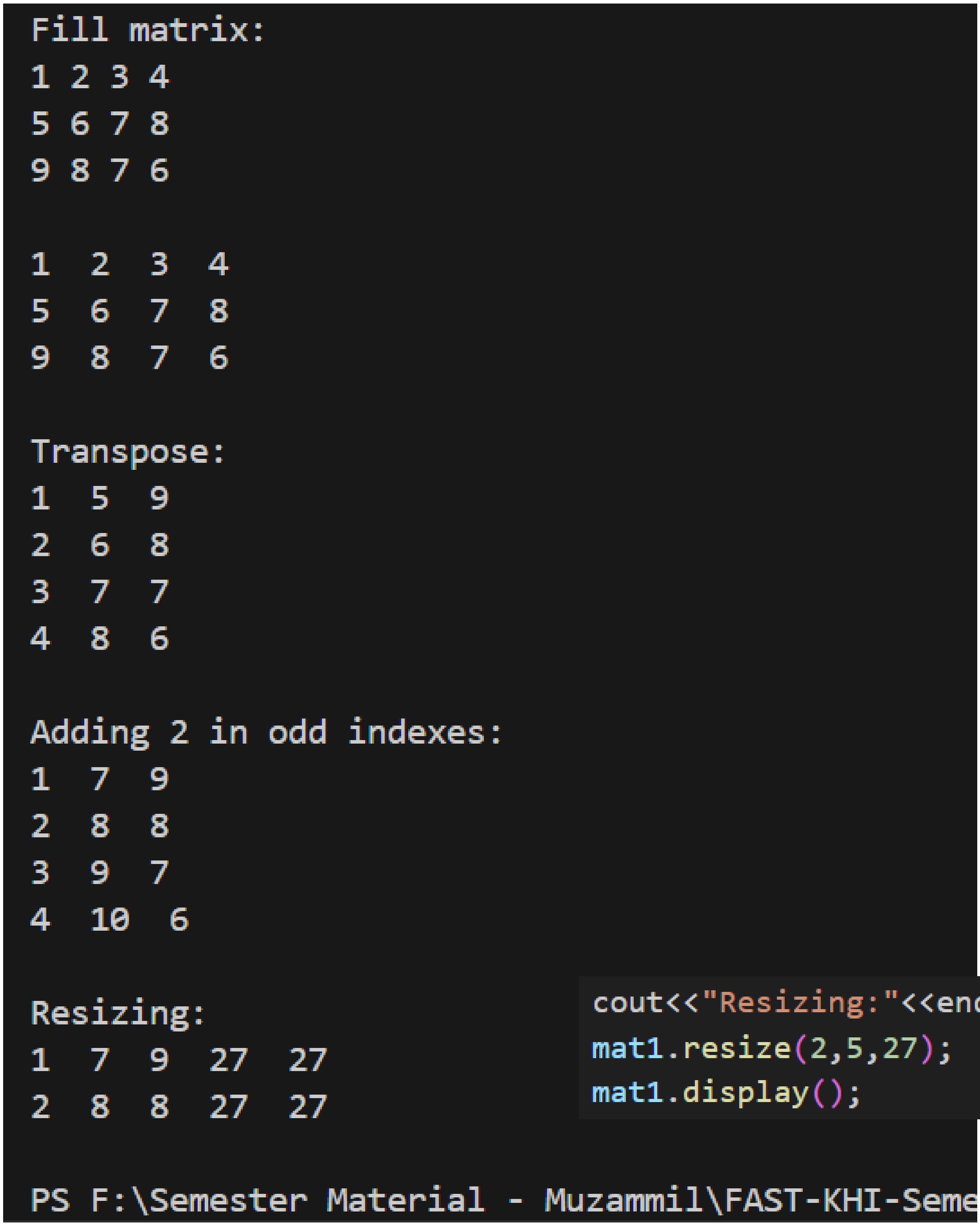


```
int main(){
    dynamic mat1(3,4);
    cout<<"Fill matrix:"<<endl;
    mat1.fill();

    cout<<"Transpose: "<<endl;
    mat1.transpose();
    mat1.display();

    cout<<"Adding 2 in odd indexes:"<<endl;
    mat1.Add2Odd();
    mat1.display();

    cout<<"Resizing:"<<endl;
    mat1.resize(3,2);
    mat1.display();
    return 0;
}
```

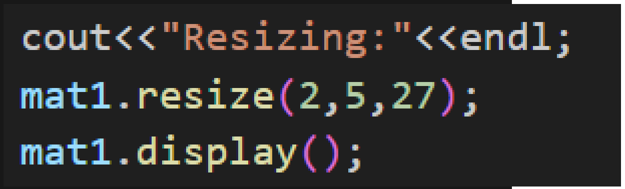


```
Fill matrix:
1 2 3 4
5 6 7 8
9 8 7 6

1  2  3  4
5  6  7  8
9  8  7  6

Transpose:
1  5  9
2  6  8
3  7  7
4  8  6

Adding 2 in odd indexes:
1  7  9
2  8  8
3  9  7
4  10  6

Resizing:
1  7
2  8
3  9

PS F:\Semester Material - Muzammil\FAST-KHI-S
```

```
cout<<"Resizing:"<<endl;
mat1.resize(3,2);
mat1.display();
```



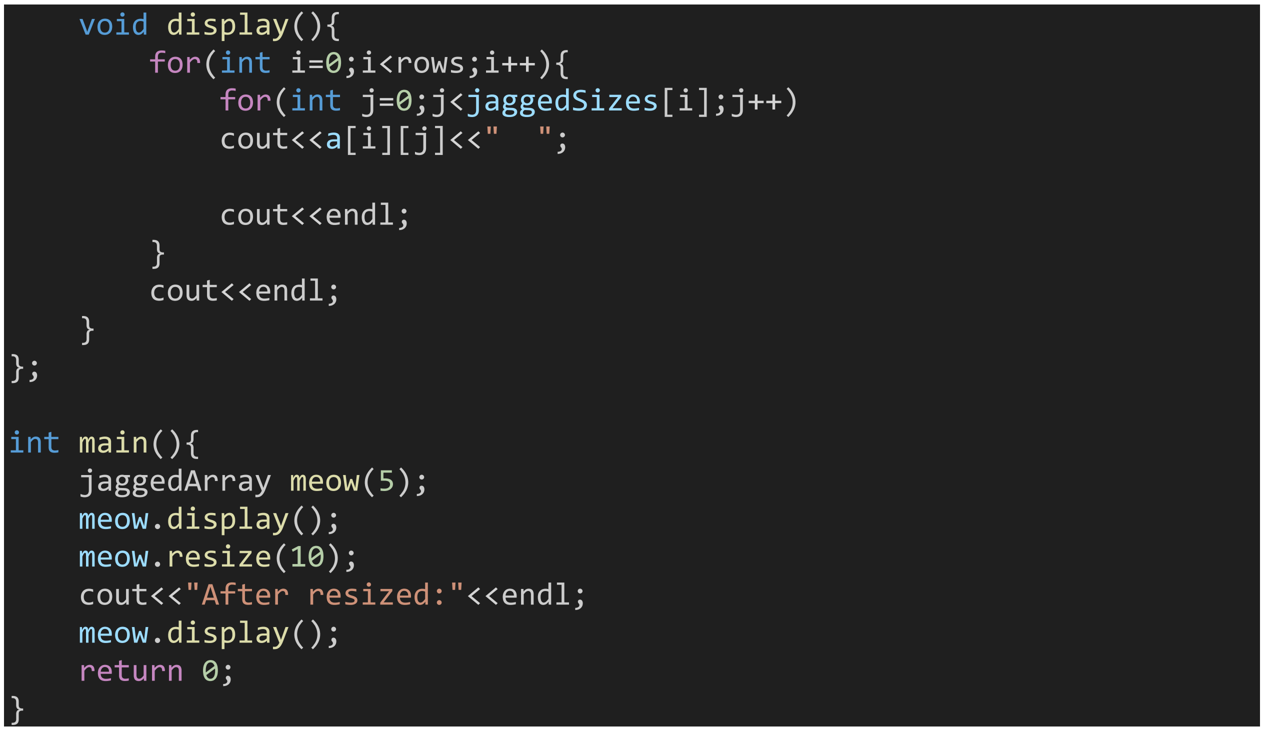
```
Fill matrix:
1 2 3 4
5 6 7 8
9 8 7 6

1  2  3  4
5  6  7  8
9  8  7  6

Transpose:
1  5  9
2  6  8
3  7  7
4  8  6

Adding 2 in odd indexes:
1  7  9
2  8  8
3  9  7
4  10  6

Resizing:
1  7  9  27  27
2  8  8  27  27

PS F:\Semester Material - Muzammil\FAST-KHI-Seme
```



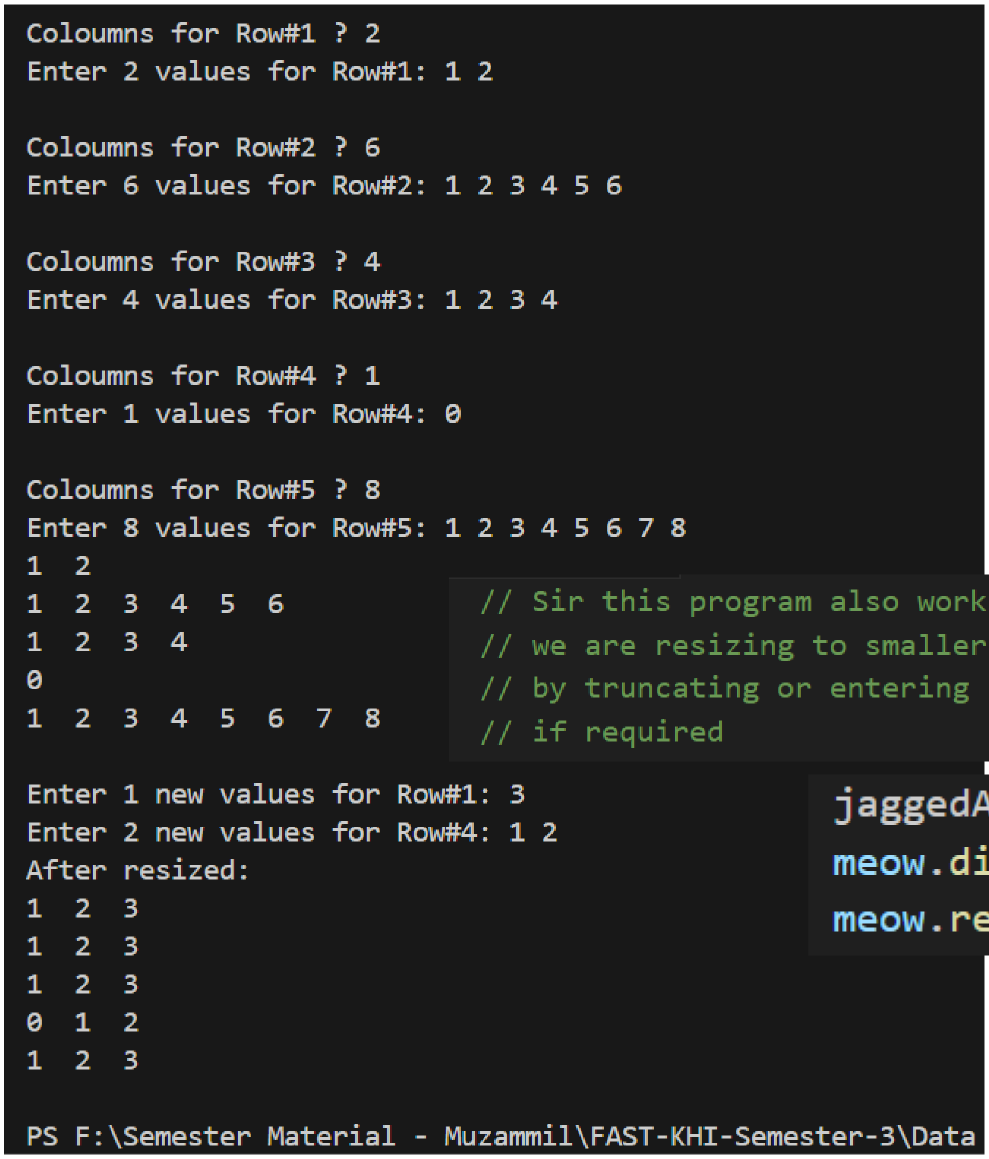
```
cout<<"Resizing:"<<endl;
mat1.resize(2,5,27);
mat1.display();
```

Q2:

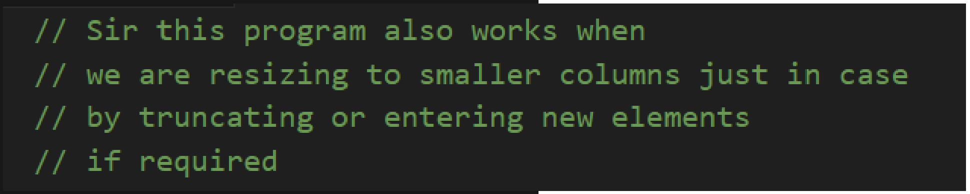
```cpp
//23K2001 - Muzammil
#include<iostream>
using namespace std;

class jaggedArray{
    private:
    int **a;
    int *jaggedSizes;
    int rows;

    public:
    jaggedArray(){a=NULL;
    jaggedSizes=NULL;
    rows=0; }
    ~jaggedArray(){
        for(int i=0;i<rows;i++)
            delete[] a[i];
        delete[] a;
        delete[] jaggedSizes;
    }

    jaggedArray(int r){
        rows = r;
        a = new int*[rows];
        jaggedSizes = new int[rows];

        int c;
        for(int i=0;i<rows;i++){
            cout<<"\nColoumns for Row#"<<i+1<<" ? ";
            cin>>c;
            a[i] = new int[c];
            jaggedSizes[i]=c;
            cout<<"Enter "<<c<<" values for Row#"<<i+1<<":
";

            for(int j=0;j<c;j++)
                cin>>a[i][j];
        }
    }

    void resize(int c){
        int **old = new int*[rows];
        int *oldSizes = new int[rows];
        for(int i=0;i<rows;i++){
            old[i] = new int[jaggedSizes[i]];
            oldSizes[i] = jaggedSizes[i];
            for(int j=0;j<jaggedSizes[i];j++)
            old[i][j] = a[i][j];
        }

        for(int i=0;i<rows;i++)
            delete[] a[i];
        delete[] jaggedSizes;
        jaggedSizes = new int[rows];

        for(int i=0;i<rows;i++){
            a[i] = new int[c];
            jaggedSizes[i]=c;
            for(int j=0;j<(oldSizes[i] < c ? oldSizes[i]:
c);j++)
            a[i][j] = old[i][j];

            if(c>oldSizes[i]){
            cout<<"Enter "<<c-oldSizes[i]<<" new values for
Row#"<<i+1<<": ";
            for(int j=oldSizes[i];j<c;j++)
                cin>>a[i][j];
            }
        }

        for(int i=0;i<rows;i++)
            delete[] old[i];
        delete[] old;
        delete[] oldSizes;
    }
```



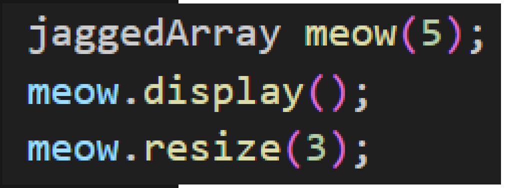
```cpp
    void display(){
        for(int i=0;i<rows;i++){
            for(int j=0;j<jaggedSizes[i];j++)
            cout<<a[i][j]<<"  ";

            cout<<endl;
        }
        cout<<endl;
    }
};

int main(){
    jaggedArray meow(5);
    meow.display();
    meow.resize(10);
    cout<<"After resized:"<<endl;
    meow.display();
    return 0;
}
```



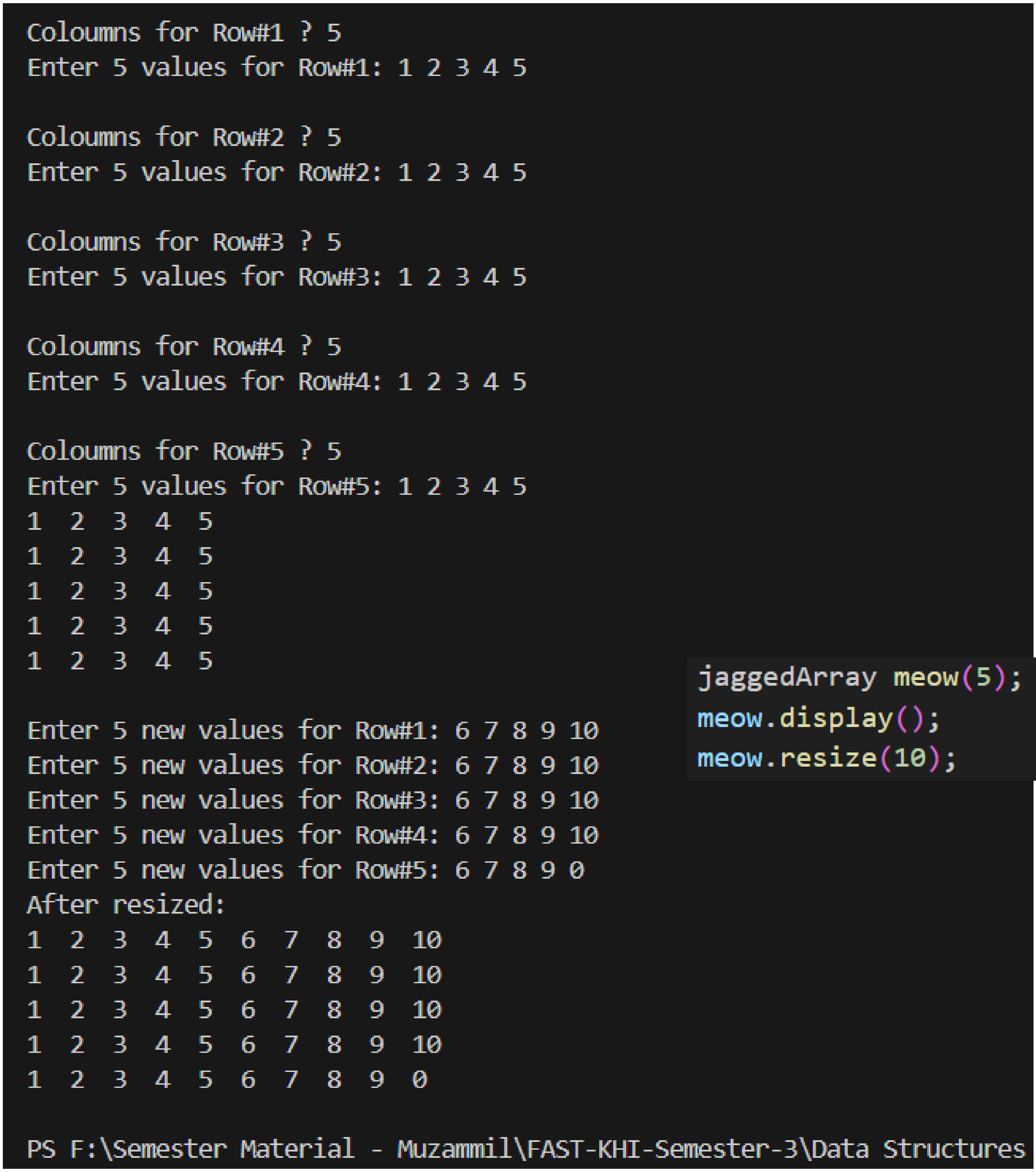
```
Coloumns for Row#1 ? 2
Enter 2 values for Row#1: 1 2

Coloumns for Row#2 ? 6
Enter 6 values for Row#2: 1 2 3 4 5 6

Coloumns for Row#3 ? 4
Enter 4 values for Row#3: 1 2 3 4

Coloumns for Row#4 ? 1
Enter 1 values for Row#4: 0

Coloumns for Row#5 ? 8
Enter 8 values for Row#5: 1 2 3 4 5 6 7 8
1  2
1  2  3  4  5  6
1  2  3  4
0
1  2  3  4  5  6  7  8

Enter 1 new values for Row#1: 3
Enter 2 new values for Row#4: 1 2
After resized:
1  2  3
1  2  3
1  2  3
0  1  2
1  2  3

PS F:\Semester Material - Muzammil\FAST-KHI-Semester-3\Data
```



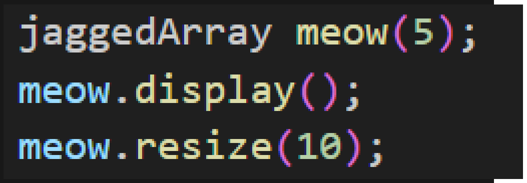
```cpp
// Sir this program also works when
// we are resizing to smaller columns just in case
// by truncating or entering new elements
// if required
```



```cpp
jaggedArray meow(5);
meow.display();
meow.resize(3);
```

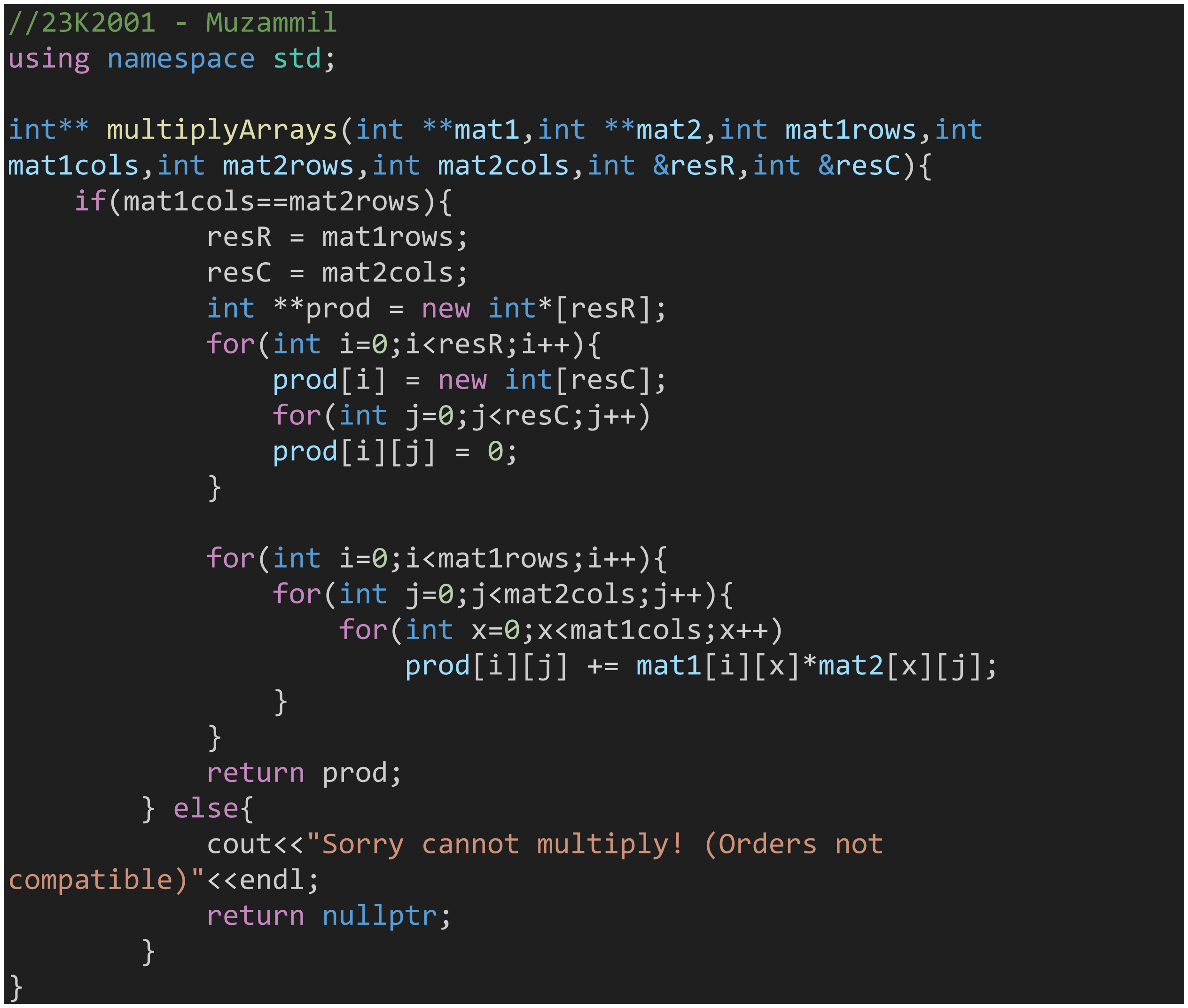


```
Coloumns for Row#1 ? 5
Enter 5 values for Row#1: 1 2 3 4 5

Coloumns for Row#2 ? 5
Enter 5 values for Row#2: 1 2 3 4 5

Coloumns for Row#3 ? 5
Enter 5 values for Row#3: 1 2 3 4 5

Coloumns for Row#4 ? 5
Enter 5 values for Row#4: 1 2 3 4 5

Coloumns for Row#5 ? 5
Enter 5 values for Row#5: 1 2 3 4 5
1  2  3  4  5
1  2  3  4  5
1  2  3  4  5
1  2  3  4  5
1  2  3  4  5

Enter 5 new values for Row#1: 6 7 8 9 10
Enter 5 new values for Row#2: 6 7 8 9 10
Enter 5 new values for Row#3: 6 7 8 9 10
Enter 5 new values for Row#4: 6 7 8 9 10
Enter 5 new values for Row#5: 6 7 8 9 0
After resized:
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  0

PS F:\Semester Material - Muzammil\FAST-KHI-Semester-3\Data Structures
```

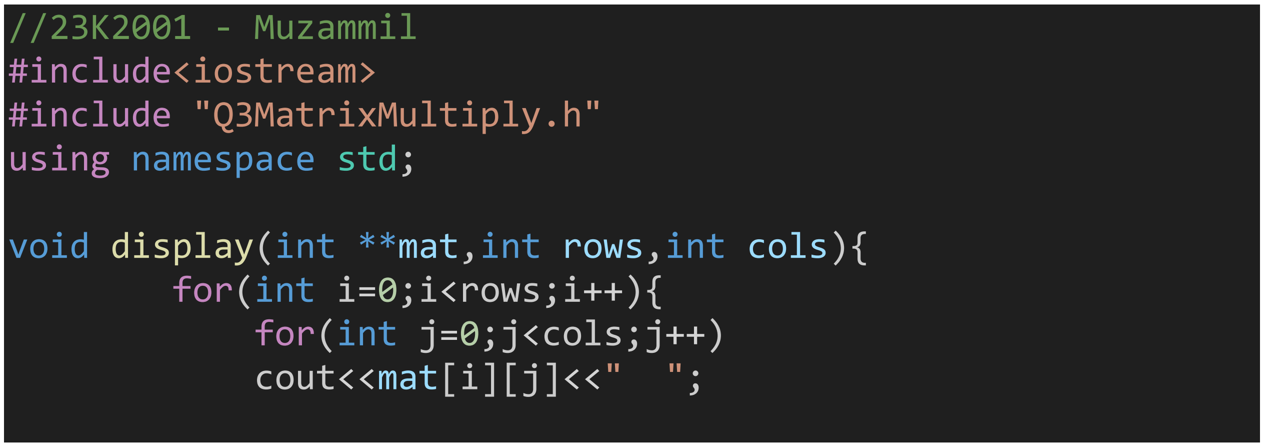


```
jaggedArray meow(5);
meow.display();
meow.resize(10);
```

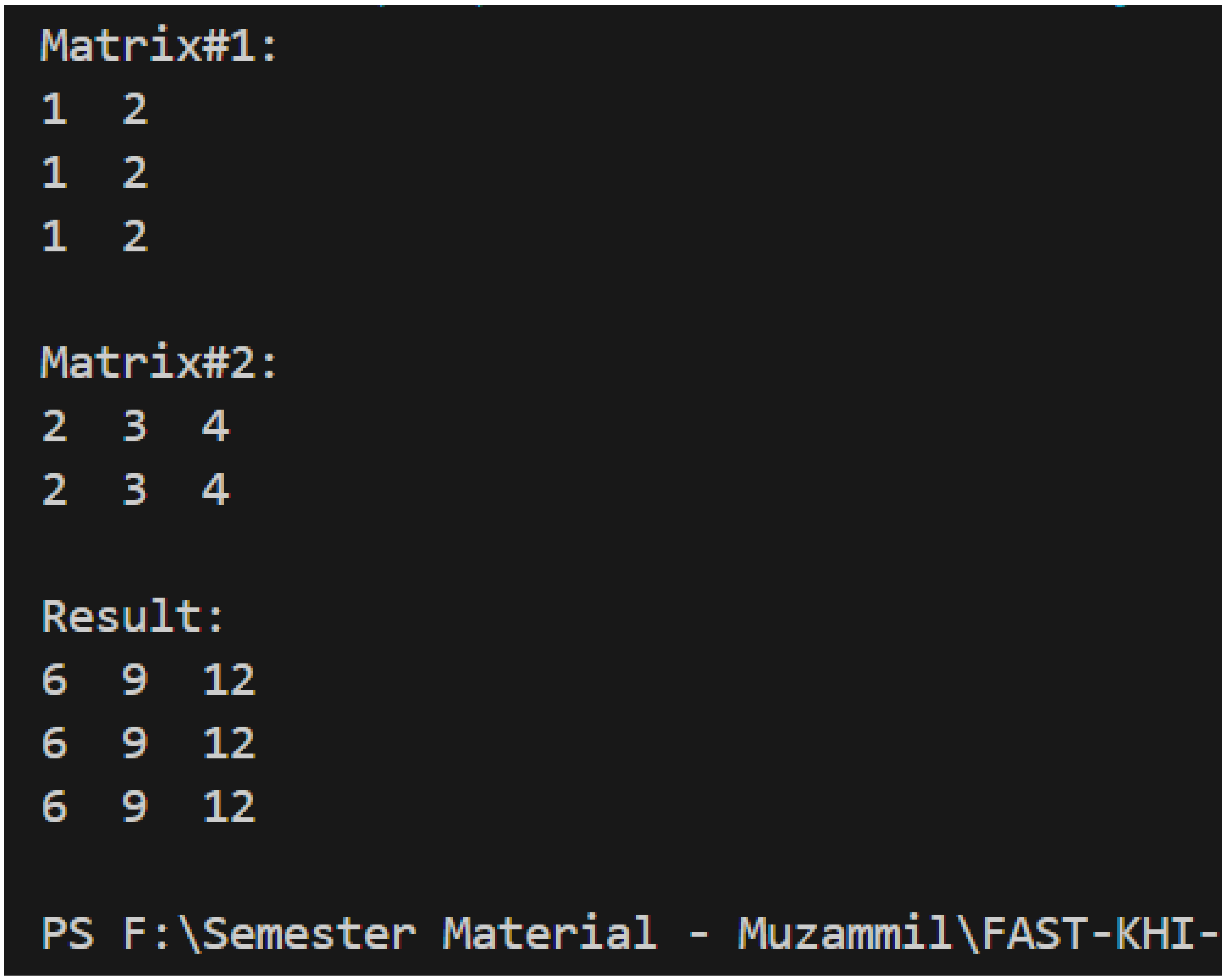
Q3:



```cpp
//23K2001 - Muzammil
using namespace std;

int** multiplyArrays(int **mat1,int **mat2,int mat1rows,int
mat1cols,int mat2rows,int mat2cols,int &resR,int &resC){
    if(mat1cols==mat2rows){
            resR = mat1rows;
            resC = mat2cols;
            int **prod = new int*[resR];
            for(int i=0;i<resR;i++){
                prod[i] = new int[resC];
                for(int j=0;j<resC;j++)
                prod[i][j] = 0;
            }

            for(int i=0;i<mat1rows;i++){
                for(int j=0;j<mat2cols;j++){
                    for(int x=0;x<mat1cols;x++)
                        prod[i][j] += mat1[i][x]*mat2[x][j];
                }
            }
            return prod;
        } else{
            cout<<"Sorry cannot multiply! (Orders not
compatible)"<<endl;
            return nullptr;
        }
}
```

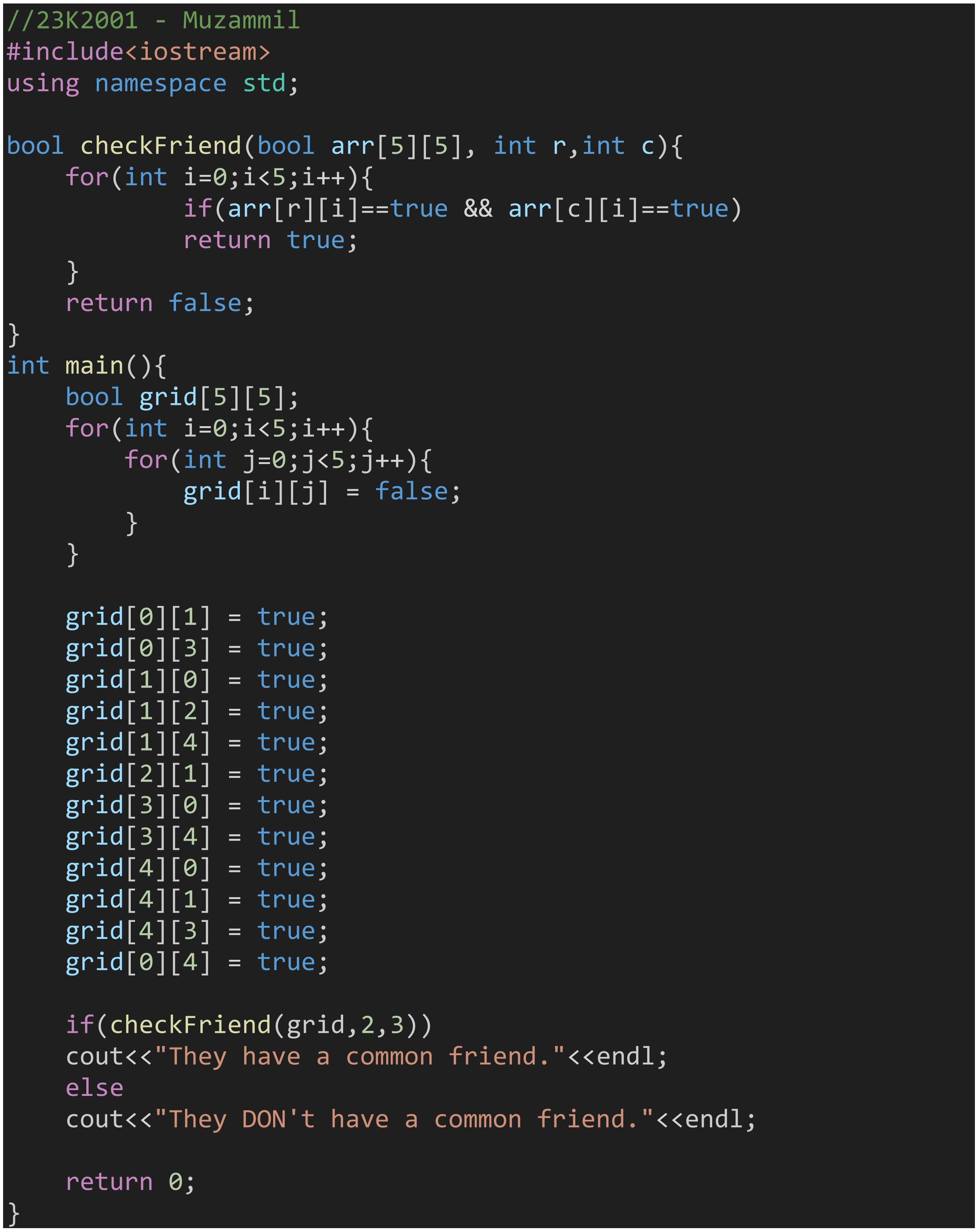


```cpp
//23K2001 - Muzammil
#include<iostream>
#include "Q3MatrixMultiply.h"
using namespace std;

void display(int **mat,int rows,int cols){
        for(int i=0;i<rows;i++){
            for(int j=0;j<cols;j++)
            cout<<mat[i][j]<<"  ";
            cout<<endl;
        }
        cout<<endl;
    }

int main(){
    int **M = new int*[3];
    for(int i=0;i<3;i++)
    M[i] = new int[2];

    int **N = new int*[2];
    for(int i=0;i<2;i++)
    N[i] = new int[3];

    for(int i=0;i<3;i++)
        for(int j=0;j<2;j++)
            M[i][j] = j+1;

    for(int i=0;i<2;i++)
        for(int j=0;j<3;j++)
            N[i][j] = j+2;

    int r = 0, c = 0;
    int **output = multiplyArrays(M,N,3,2,2,3,r,c);

    cout<<"Matrix#1:"<<endl;
    display(M,3,2);
    cout<<"Matrix#2:"<<endl;
    display(N,2,3);
    cout<<"Result:"<<endl;
    display(output,r,c);

    for(int i=0;i<3;i++)
        delete[] M[i];
    delete[] M;

    for(int i=0;i<2;i++)
        delete[] N[i];
    delete[] N;
    for(int i=0;i<r;i++)
        delete[] output[i];
    delete[] output;

    return 0;
}
```

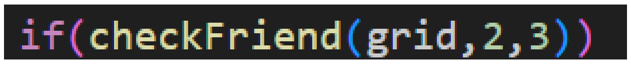


```
Matrix#1:
1  2
1  2
1  2

Matrix#2:
2  3  4
2  3  4

Result:
6  9  12
6  9  12
6  9  12

PS F:\Semester Material - Muzammil\FAST-KHI-
```

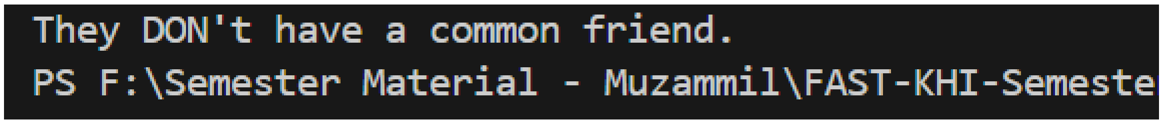
Q4:



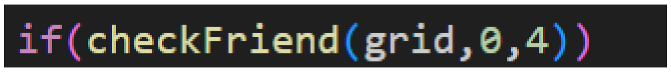
```cpp
//23K2001 - Muzammil
#include<iostream>
using namespace std;

bool checkFriend(bool arr[5][5], int r,int c){
    for(int i=0;i<5;i++){
            if(arr[r][i]==true && arr[c][i]==true)
            return true;
    }
    return false;
}
int main(){
    bool grid[5][5];
    for(int i=0;i<5;i++){
        for(int j=0;j<5;j++){
            grid[i][j] = false;
        }
    }

    grid[0][1] = true;
    grid[0][3] = true;
    grid[1][0] = true;
    grid[1][2] = true;
    grid[1][4] = true;
    grid[2][1] = true;
    grid[3][0] = true;
    grid[3][4] = true;
    grid[4][0] = true;
    grid[4][1] = true;
    grid[4][3] = true;
    grid[0][4] = true;

    if(checkFriend(grid,2,3))
    cout<<"They have a common friend."<<endl;
    else
    cout<<"They DON't have a common friend."<<endl;

    return 0;
}
```



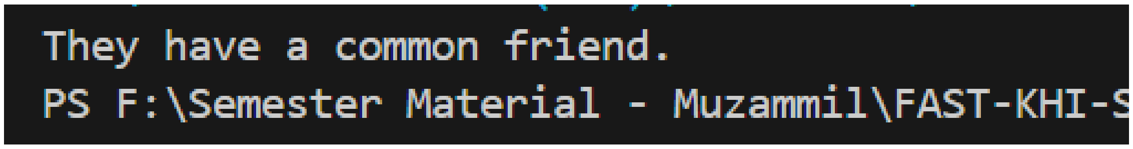
```
if(checkFriend(grid,2,3))
```



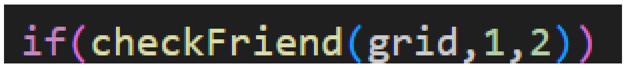
```
They DON't have a common friend.
PS F:\Semester Material - Muzammil\FAST-KHI-Semeste
```



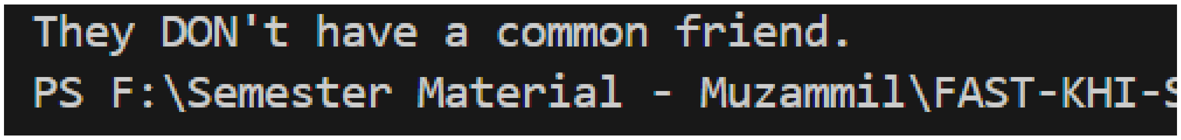
```
if(checkFriend(grid,0,4))
```



```
They have a common friend.
PS F:\Semester Material - Muzammil\FAST-KHI-S
```

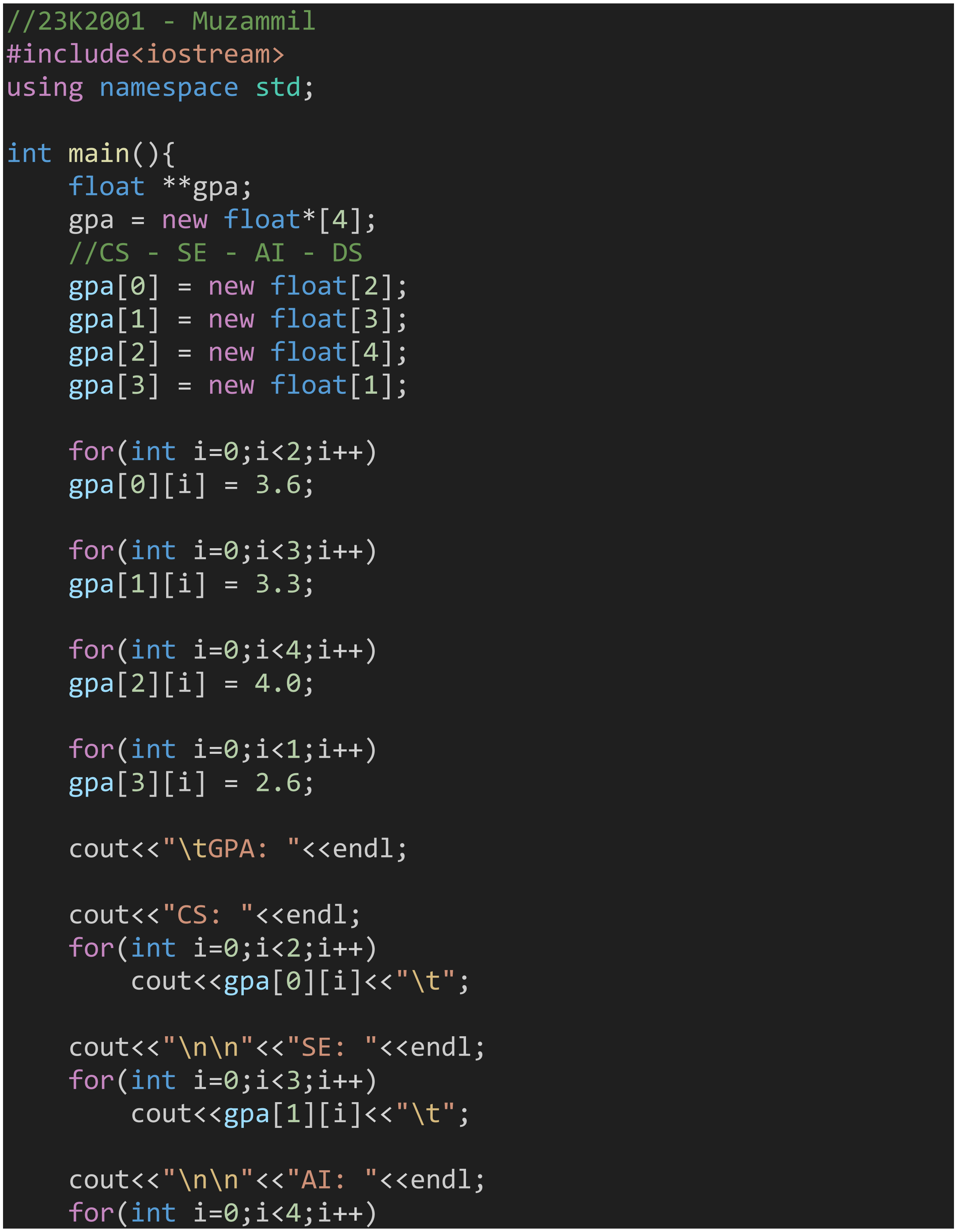


```
if(checkFriend(grid,1,2))
```

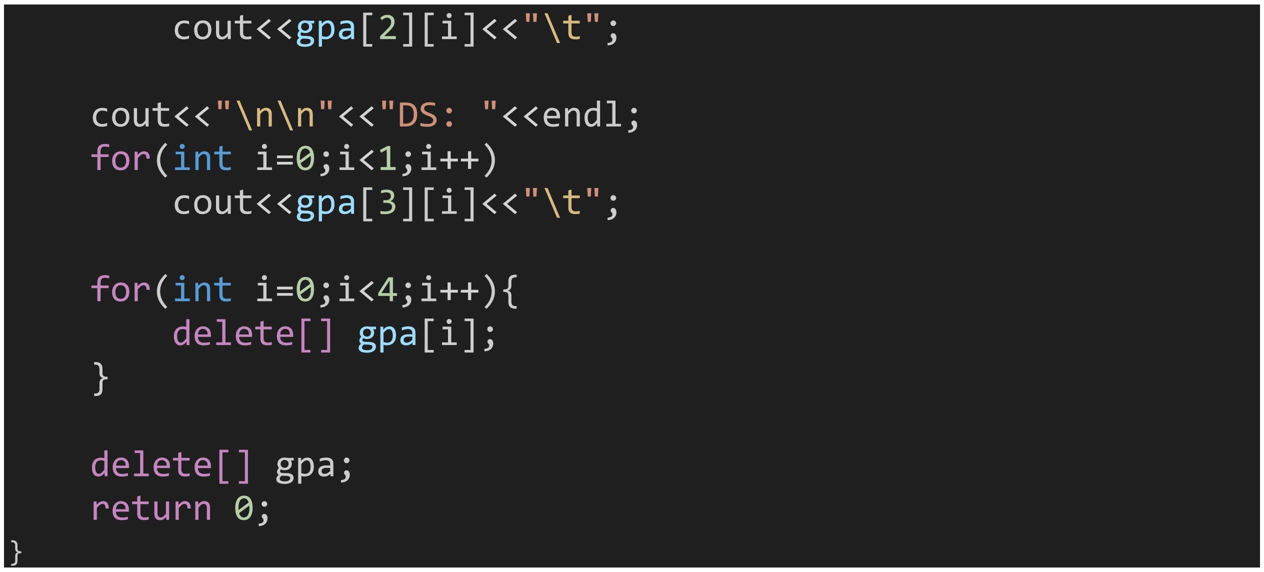


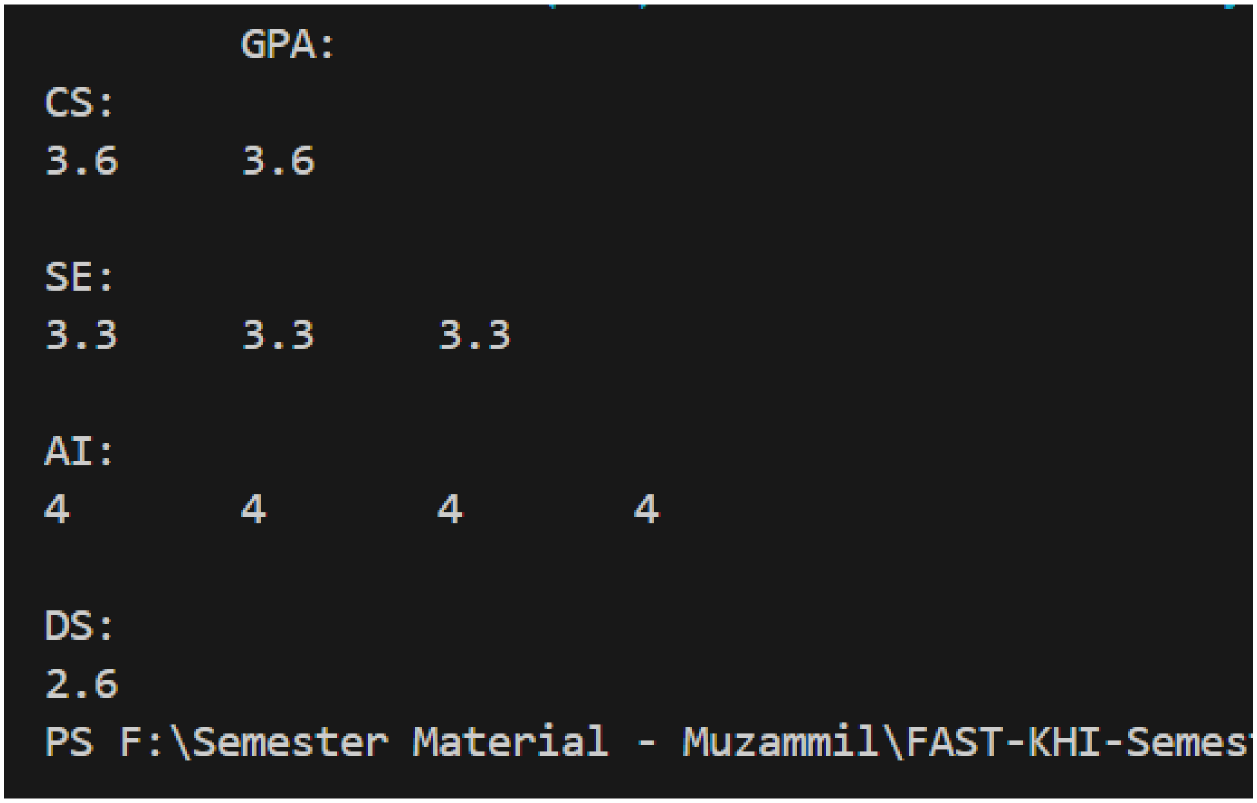
```
They DON't have a common friend.
PS F:\Semester Material - Muzammil\FAST-KHI-S
```

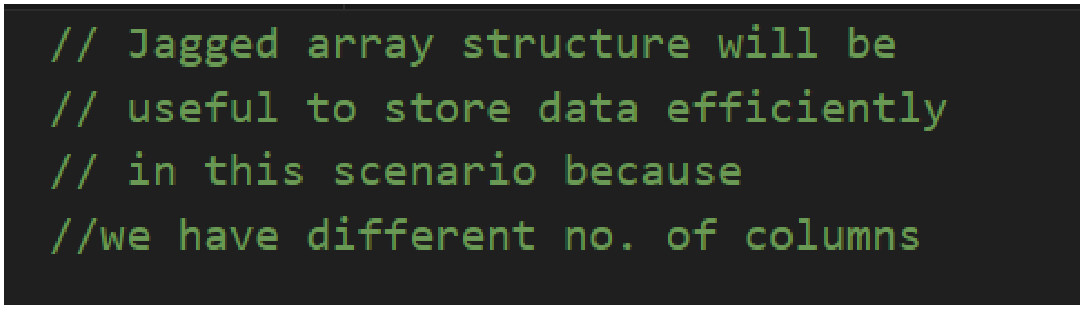
Q5:



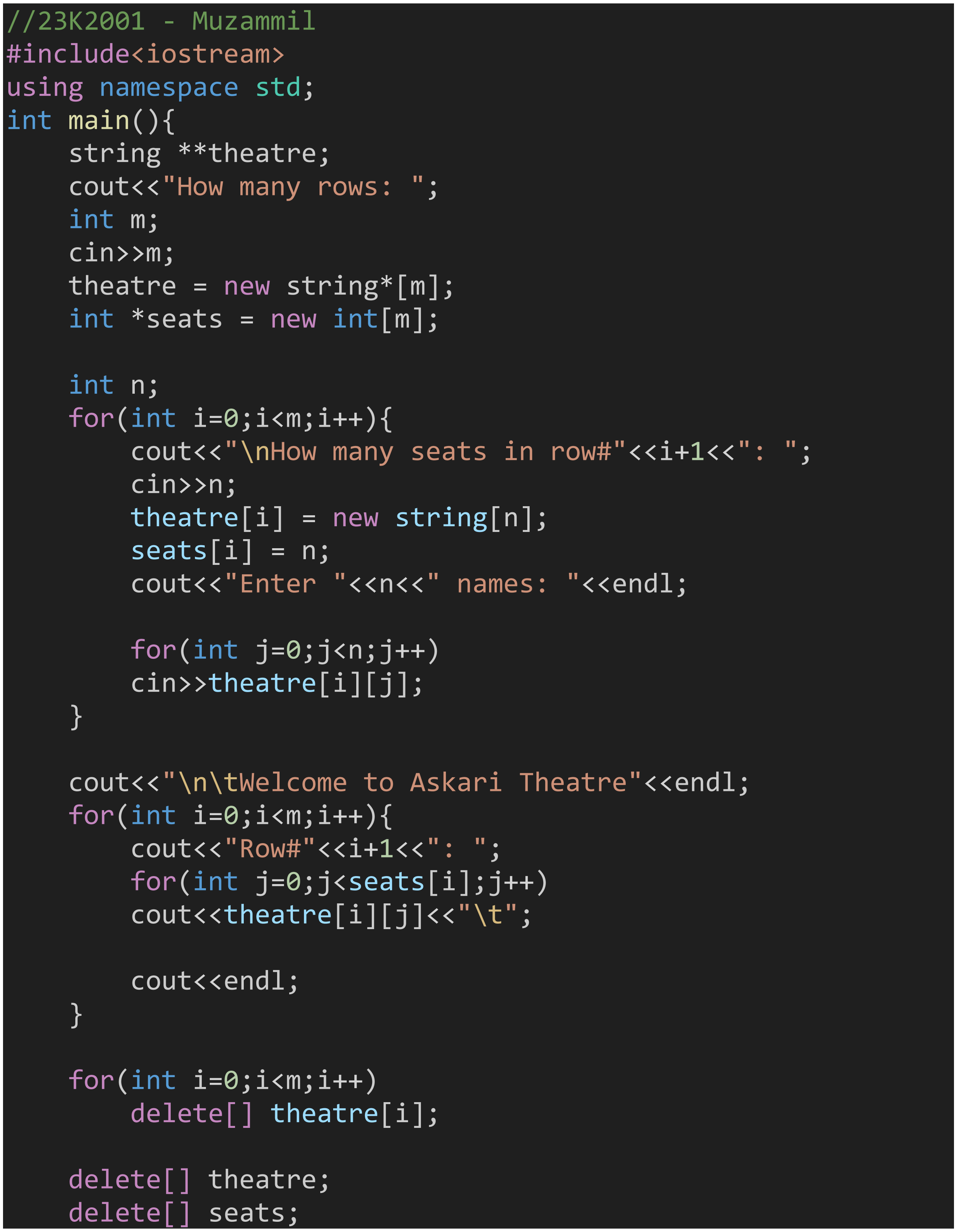
```
//23K2001 - Muzammil
#include<iostream>
using namespace std;

int main(){
    float **gpa;
    gpa = new float*[4];
    //CS - SE - AI - DS
    gpa[0] = new float[2];
    gpa[1] = new float[3];
    gpa[2] = new float[4];
    gpa[3] = new float[1];

    for(int i=0;i<2;i++)
    gpa[0][i] = 3.6;

    for(int i=0;i<3;i++)
    gpa[1][i] = 3.3;

    for(int i=0;i<4;i++)
    gpa[2][i] = 4.0;

    for(int i=0;i<1;i++)
    gpa[3][i] = 2.6;

    cout<<"\tGPA: "<<endl;

    cout<<"CS: "<<endl;
    for(int i=0;i<2;i++)
        cout<<gpa[0][i]<<"\t";

    cout<<"\n\n"<<"SE: "<<endl;
    for(int i=0;i<3;i++)
        cout<<gpa[1][i]<<"\t";

    cout<<"\n\n"<<"AI: "<<endl;
    for(int i=0;i<4;i++)
```


```
        cout<<gpa[2][i]<<"\t";

    cout<<"\n\n"<<"DS: "<<endl;
    for(int i=0;i<1;i++)
        cout<<gpa[3][i]<<"\t";

    for(int i=0;i<4;i++){
        delete[] gpa[i];
    }

    delete[] gpa;
    return 0;
}
```

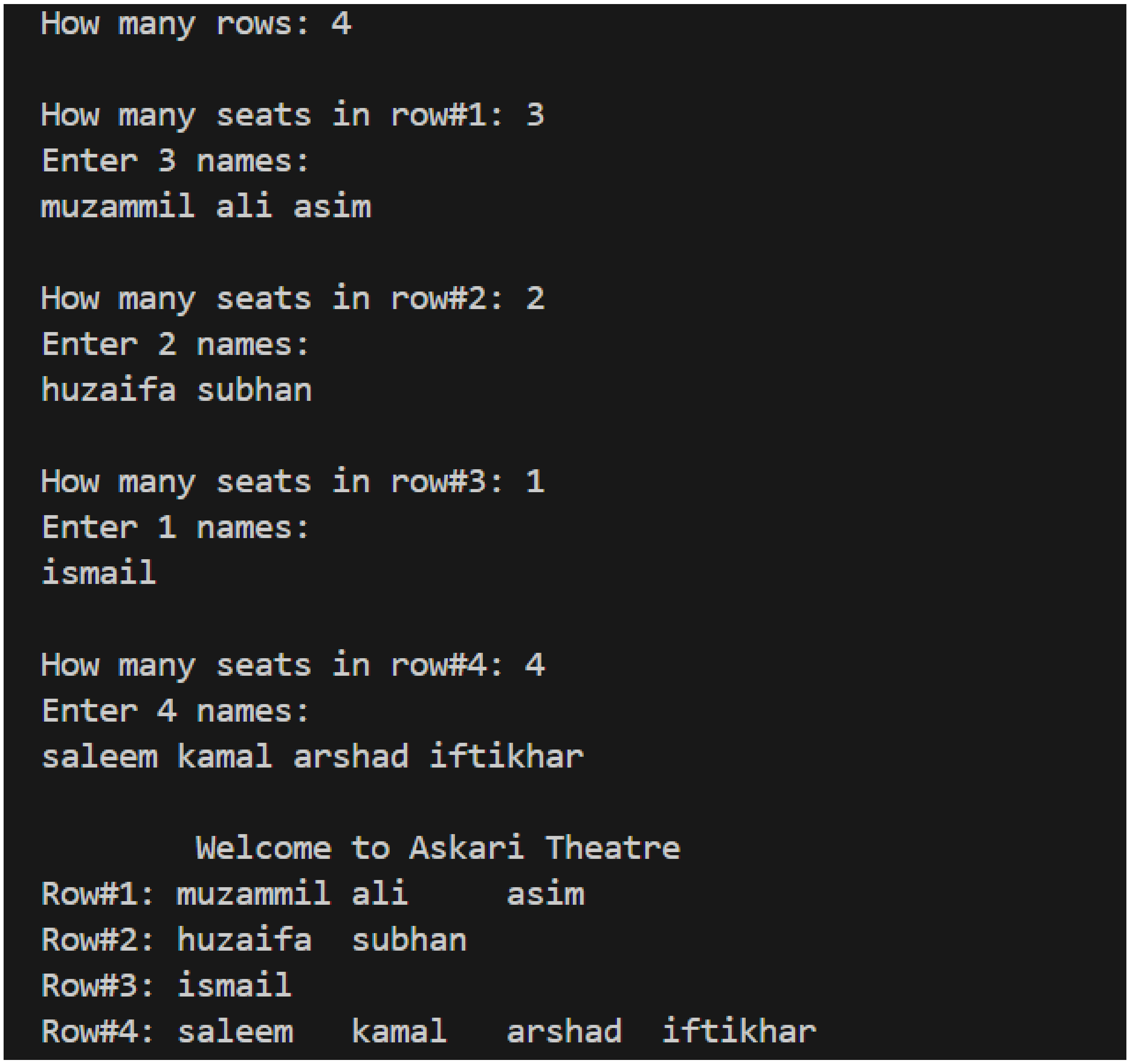

```
        GPA:
CS:
3.6     3.6

SE:
3.3     3.3     3.3

AI:
4       4       4       4

DS:
2.6
PS F:\Semester Material - Muzammil\FAST-KHI-Semes
```


```
// Jagged array structure will be
// useful to store data efficiently
// in this scenario because
//we have different no. of columns
```

Q6:



```cpp
//23K2001 - Muzammil
#include<iostream>
using namespace std;
int main(){
    string **theatre;
    cout<<"How many rows: ";
    int m;
    cin>>m;
    theatre = new string*[m];
    int *seats = new int[m];

    int n;
    for(int i=0;i<m;i++){
        cout<<"\nHow many seats in row#"<<i+1<<": ";
        cin>>n;
        theatre[i] = new string[n];
        seats[i] = n;
        cout<<"Enter "<<n<<" names: "<<endl;

        for(int j=0;j<n;j++)
        cin>>theatre[i][j];
    }

    cout<<"\n\tWelcome to Askari Theatre"<<endl;
    for(int i=0;i<m;i++){
        cout<<"Row#"<<i+1<<": ";
        for(int j=0;j<seats[i];j++)
        cout<<theatre[i][j]<<"\t";

        cout<<endl;
    }

    for(int i=0;i<m;i++)
        delete[] theatre[i];

    delete[] theatre;
    delete[] seats;
    return 0;
}
```



```
How many rows: 4

How many seats in row#1: 3
Enter 3 names:
muzammil ali asim

How many seats in row#2: 2
Enter 2 names:
huzaifa subhan

How many seats in row#3: 1
Enter 1 names:
ismail

How many seats in row#4: 4
Enter 4 names:
saleem kamal arshad iftikhar

        Welcome to Askari Theatre
Row#1: muzammil ali     asim
Row#2: huzaifa  subhan
Row#3: ismail
Row#4: saleem   kamal   arshad  iftikhar
```