DSA Lab02

23K2001

M.Muzammil Siddiqui

BCS-3J

Practice Task:

//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;

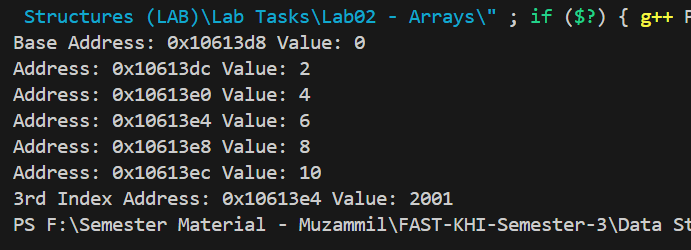
    num[3] = 2001;

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

    delete[] num;

    return 0;

}



Q1:

//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();

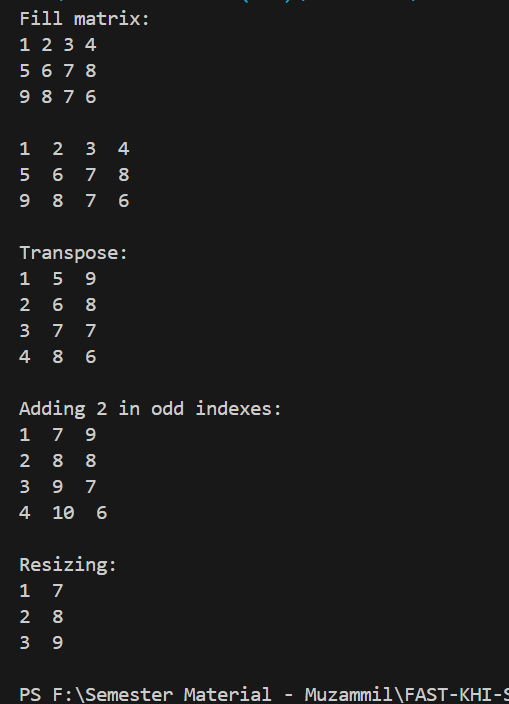
    cout<<"Resizing:"<<endl;

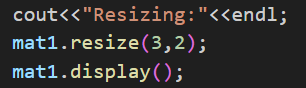
    mat1.resize(3,2);

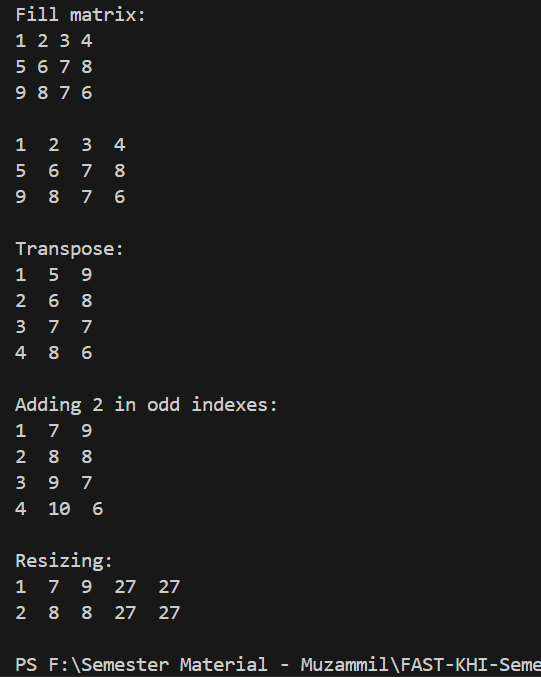
    mat1.display();

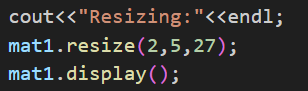
    return 0;

}









Q2:

//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;

    }

    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();

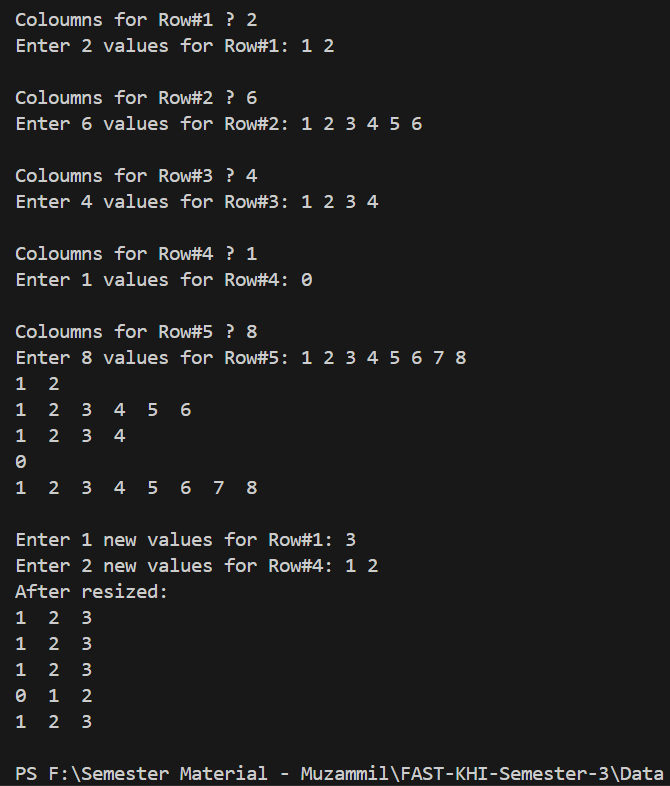
    meow.resize(10);

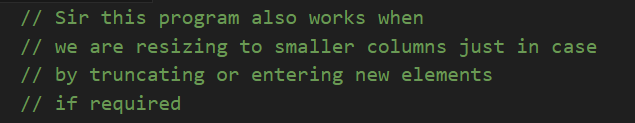
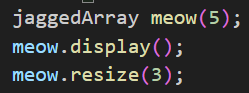
cout<<"After resized:"<<endl;

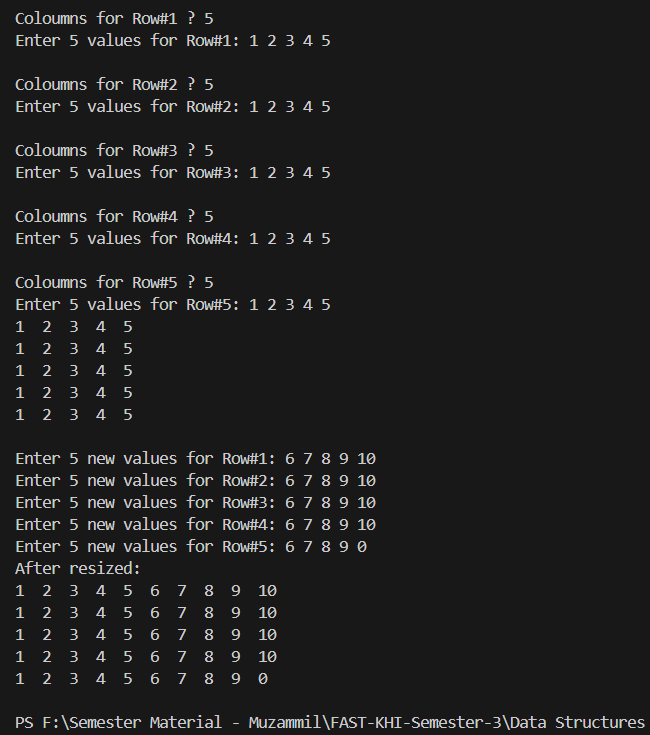
    meow.display();

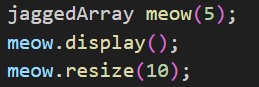
    return 0;

}









Q3:

//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;

        }

}

//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;

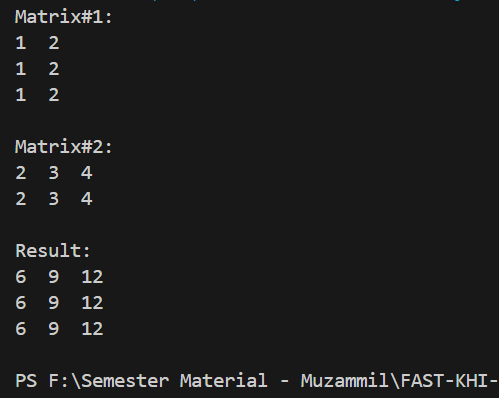
    for(int i=0;i<r;i++)

        delete[] output[i];

    delete[] output;

    return 0;

}



Q4:

//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;

}

A black background with white text

Description automatically generated

A black background with white text

Description automatically generated

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];

    }

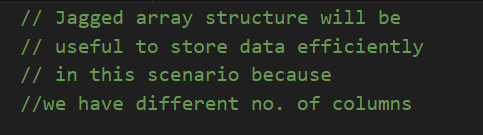
    delete[] gpa;

    return 0;

}

A screen shot of a computer

Description automatically generated



Q6:

//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;

}

