

```

#include <iostream.h>

void add(double **,double **, int, int);
void sum(double *,double*, int, int);
void sum2(double [][][4], double [][][4], int);

int main()
{
/* Creating 2-dim. arrays: 3 choices */
    const int nrows = 5;
    const int ncols = 4;

    //A on stack and heap
    double* A[nrows];

    for (int i =0; i<nrows; i++)
        A[i] = new double[ncols];

    //B on heap
    double **B = new double*[nrows];
    for (i =0; i<nrows; i++)
        B[i] = new double[ncols];

    //C on stack
    double C[nrows][ncols] = {{0,0}};

    A[0][0] = 1;
    **A = 2;
    *A[1] = 3;
    *(A[1]) = 4;
    (*A)[1] = 5;

    B[0][0] = 1;
    **B = 2;
    *B[0] = 3;

    C[0][0] = 1;
    **C = 2;
    *C[0] = 3;

    for(i=0; i<nrows; i++)
        for(int j=0; j < ncols; j++)
        {
            A[i][j] = i*j;
            B[i][j] = i*j;
            C[i][j] = i*j;
        }

    add(B,B,nrows,ncols);
    add(A,A,nrows,ncols);

    cout << "A is: " << endl;
    for(i=0; i<nrows; i++)
    {
        for(int j=0; j < ncols; j++)
            cout << A[i][j] << '\t';
    }
}

```

```

        cout << endl;
    }

    cout << "B is: " << endl;
    for(i=0; i<nrows; i++)
    {
        for(int j=0; j < ncols; j++)
            cout << B[i][j] << '\t';
        cout << endl;
    }

//    add(C,C,2,2); //doesn't work
//One solution:
sum(C[0],C[0],nrows,ncols);
//    sum(&C[0][0],&C[0][0],2,2);
cout << "C is: " << endl;
for(i=0; i<nrows; i++)
{
    for(int j=0; j < ncols; j++)
        cout << C[i][j] << '\t';
    cout << endl;
}

//Another solution:
sum2(C,C,nrows);

//deleting A
for(i=0; i < nrows; i++)
    delete [] A[i];

//delete B
for(i=0; i< ncols; i++)
    delete B[i];
delete [] B;

return 0;
}

void sum(double *A, double *B,int nrows, int ncols)
{
    for(int i =0; i< nrows*ncols; i++)
        B[i] += A[i];
    return;
}

void add(double **A, double **B,int nrows, int ncols)
{
    for(int i =0; i< nrows; i++)
        for(int j=0; j <ncols; j++)
            B[i][j] += A[i][j];
    return;
}

void sum2(double A[][4], double B[][4], int nrows)
{
    for(int i =0; i< nrows; i++)

```

```
        for(int j=0; j <4; j++)
            B[i][j] += A[i][j];
    return;
}
```