

```

#include<stdio.h>
#include<stdlib.h>
#include<time.h>
#include<string.h>
typedef void V;
typedef int I;
typedef int** A;
typedef int D[2];

V PopulateArray2DUnique(A,D);
V DisplayArray2D(A,D);
I FindLargest(A,D);
I FindColSum(A,D);
I Sort2DArray(A,D);
I CopyArray2D(A,A,D);
I CopyArray2DSpiral(A,A,D);

I main() {
    D d={4,5},dt;
    I i,j;
    A a=(I**)malloc(d[0]*sizeof(I*));
    A b=(I**)malloc(d[0]*sizeof(I*));
    srand(time(NULL));
    for(i=0;i<d[0];++i)a[i]=(I*)malloc(d[1]*sizeof(I));
    for(i=0;i<d[0];++i){b[i]=(I*)malloc(d[1]*sizeof(I));
        for(j=0;j<d[1];++j)b[i][j]=rand()%10;}
    printf("PopulateArray2DUnique a");PopulateArray2DUnique(a,d);
    printf("DisplayArray2D a\n");DisplayArray2D(a,d);
    printf("FindLargest a\n%d\n\n",FindLargest(a,d));
    dt[0]=d[0];dt[1]=1;
    printf("FindColSum 1 a\n%d\n\n",FindColSum(a,dt));
    printf("DisplayArray2D b\n");DisplayArray2D(b,d);
    Sort2DArray(b,d);
    printf("Sort2dArray b\n");DisplayArray2D(b,d);
    printf("CopyArray2D b a\n\n");CopyArray2D(/*very*/b,a,d/*style:(*/);
    printf("DisplayArray2D a\n");DisplayArray2D(a,d);
    printf("CopyArray2DSpiral b a\n\n");CopyArray2DSpiral(b,a,d);
    printf("DisplayArray2D a\n");DisplayArray2D(a,d);
    for(i=0;i<d[0];++i){free(a[i]);free(b[i]);}free(a);free(b);
    return 0;
}

V PopulateArray2DUnique(A a,D d) {
    I i,j,in;for(i=0;i<d[0];++i)for(j=0,in=i*d[1];j<d[1];++j)a[i][j]=in+j;
}

V DisplayArray2D(A a,D d) {
    I i,j;
    i=0;while(i<d[0]) {
        j=0;while(j<d[1]) {
            printf("%d ",a[i][j]);
            ++j;}
        printf("\n");
        ++i;}
    printf("\n");
}

I FindLargest(A a,D d) {
    I i,j,l=a[0][0];
    i=0;while(i<d[0]) {
        j=0;while(j<d[1]) {
            if(a[i][j]>l)l=a[i][j];
            ++j;}
        ++i;}
    return l;
}

I FindColSum(A a,D d) {
    I i,j,s=0;
    i=0;while(i<d[0]) {

```

```

        j=0;while(j<d[1]) {
            s+=a[i][j];
            ++j;}
        ++i;}
    return s;
}

V QuickSort(I* a,I d) {
    int i,j,p,t;if(d<2)return;
    p=a[d/2];for(i=0,j=d-1;;++i,--j){
        while(a[i]<p)++i;while(p<a[j])--j;
        if(i>=j)break;t=a[i];a[i]=a[j];a[j]=t;}
    QuickSort(a,i);QuickSort(a+j,d-i);
}

I Sort2DArray(A a,D d) {
    I i;for(i=0;i<d[0];++i)QuickSort(a[i],d[1]);return rand();// :D
}

I CopyArray2D(A a,A b,D d) {
    I i;for(i=0;i<d[0];++i)memcpy(b[i],a[i],d[1]*sizeof(I));return rand();
}

I CopyArray2DSpiral(A a,A b,D t) {
    I i,x=0,k=0,l=0,m=t[0],n=t[1];
    while (k < m && l < n) {
        for (i = l; i < n; ++i,++x) {
            b[x/t[1]][x%t[1]] = a[k][i];
        }
        k++;
        for (i = k; i < m; ++i,++x) {
            b[x/t[1]][x%t[1]] = a[i][n-1];
        }
        n--;
        if (k < m) {
            for (i = n-1; i >= l; --i,++x) {
                b[x/t[1]][x%t[1]] = a[m-1][i];
            }
            m--;
        }
        if (l < n) {
            for (i = m-1; i >= k; --i,++x) {
                b[x/t[1]][x%t[1]] = a[i][l];
            }
            l++;
        }
    }
    return rand();
}

```