

اسئلة امتحانات نهائية سابقة للأستاذ فرج الشركسي

في اسئلة متكررة ف الملف وفي اسئلة فيها
اختلاف بسيط لو شفت سؤال متكرر ركز فيه مرات
فيه اختلاف بسيط

أسئلة مقالية

لو لاحظتو اي خطأ قولولنا

[لو اي حد يبي الاكواد يلقاهم يضط هنا](#)

1. أكتب دالة (MUL) تستقبل مصفوفة ثنائية (A) وأبعادها (N - M)، وتضرب العناصر الفردية في الصفوف الزوجية في 5، ثم تستدعي دالة الطباعة (PRINT(A,N,M))، مع الأخذ في الاعتبار أن الدالة لا تخرج شيئاً.

```
#include <iostream>

using namespace std;
void read2DArray(int array[][100] , int rows , int columns){
    for(int i = 0; i < rows; i++)
        for(int j = 0; j < columns; j++)
            cin >> array[i][j];
}

void calculateA (int A[][100] , int rows , int columns){
    for(int i = 0; i < rows; i++)
        for(int j = 0; j < columns; j++)
            // لو الصف زوجي و العنصر فردي
            if(i % 2 == 0 && A[i][j] % 2 != 0)
                A[i][j] *= 5;
}

void print2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++)
            cout << array[i][j] << " ";

        cout << endl;
    }
}

int main(){

    int A[100][100] ;
    int N , M;
    cin >> N >> M;
    read2DArray(A , N,M);
    calculateA(A , N , M);
    print2DArray(A , N , M);

    return 0;
}
```

```
#include <iostream>

using namespace std;
void print(int A[] , int n){

    for(int i = 0 ; i < n ; i++)
        cout << A[i] << " ";

    cout << endl;

}

void readArray(int A[] , int n){

    for(int i = 0 ; i < n ; i++)
        cin >> A[i];

}

int Trans(int A[] , int B[] , int n , int bSize){
    // النصف الاول ف بنوقفو عن نص n
    for (int i = 0 ; i < n/2 ; i++)
        // لو العنصر الحالي اصغر من 50
        if(A[i] < 50){
            // ننسخه ل B
            B[bSize] = A[i];
            // نزيدو حجم B واحد
            bSize++;
        }
    // نرجعو الحجم الجديد
    return bSize;
}

// في حالة نسخ مش نقل متدورهاش الدالة هذي
int reFormatArray(int A[] , int n){
    // نفس الكلام متاع دالة Trans مع اضافة بسيطة
    for(int i = 0 ; i < n/2 ; i++)
        if(A[i] < 50){
            for(int j = i ; j < n ; j++)
                A[j] = A[j+1];
            // تقريبا شرحتها فأسئلة النصف الفكرة
            // بنعدو كل ما ننقلو ننقصو من الطول 1 والدالة بتعاود تنادي نفسها
            // بيش تبدأ من جديد بالطول الجديد
            return reFormatArray(A , n-1);
        }
    // لو ملقتش ولا واحد يعني نقلتهم كلهم ترجع الحجم الجديد
    return n;
}

int main(){

    int n ,A[1000] , B[1000] , bSize= 0;

    cin >> n;

    readArray(A,n);
    // ننقلو العناصر من A ل B ونرجعو حجم B الجديد
    bSize = Trans(A , B, n , bSize);
    // نحولو العناصر يلي نقلناهم من A ونرجعو حجمها الجديد
    n = reFormatArray(A,n);
    // نطبعو A
    print(A,n);
    // نطبعو B
    print(B,bSize);


    return 0;
}
```

3. باستخدام المؤشرات، اكتب دالة تستقبل مصفوفة ثنائية (SEARCH) وإعادها ورقفاً صحيحاً، وتبحث عن الرقم، فإن وجدته تطبع عناصر الصف الذي فيه الرقم، وإلا تطبع: NO، مع الأخذ في الاعتبار أن الدالة لا ترجع شيئاً.



```
#include <iostream>

using namespace std;

void read2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++){
            // الكلام الجديد هنا
            // *(array + i) == array[i]
            // (*(array + i) + j) == array[i][j]
            cin >> *( *(array + i) + j);
        }
    }
}

void printCurrentRow(int array[][100] , int row , int columns ){
    for(int i = 0; i < columns; i++)
        // *( *(array + row) + i) == array[row][i]
        cout << *( *(array + row) + i) << " ";
    cout << endl;
}

void Search(int array[][100] , int rows , int columns , int x){

    for(int i = 0; i < rows; i++)
        for(int j = 0; j < columns; j++)
            // لو العنصر لحالي يساوي x
            // نطبعو الصف متاعها كامل
            if( *( *(array + i) + j) == x)
                return printCurrentRow(array , i , columns);
    // لو ملقيناهاش
    cout << "NO" ;
}

int main(){

    int A[100][100];

    int rows , cols , x ;

    cin >> rows >> cols >> x ;
    // نقرأ المصفوفة بالمؤشرات
    // ببساطة بنوصلو لعناصر المصفوفة عن طريق نقطة معينة فالمصفوفة
    // لان المصفوفات عناصرهم مواقعهم فذاكرة (stack) بينهم مسافة ثابتة ع حسب نوع متغيراتها
    // مش احسن شرح لكن لازم تكون فاهم المؤشرات (pointers) قبل متكلم فباقي الأسئلة متاعهم
    read2DArray(A,rows,cols);

    Search(A,rows,cols ,x);

    return 0;
}
```

```
#include <iostream>

using namespace std;

void read2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++){
            // *(array + i) == array[i]
            // (*(array + i) + j) == array[i][j]
            cin >> *( *(array + i) + j);
        }
    }

}

void print2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++)
        for(int j = 0; j < columns; j++)
            cout << array[i][j] << " ";

    cout << endl;

}

void print1DArray(int array[] , int columns){

    for(int i = 0; i < columns; i++)
        cout << array[i]<< " ";

    cout << endl;

}

void Throw (int array[][100] , int rows , int columns ) {

    int result[100];
    // طول المصفوفة الاحادية
    int tracker = 0;

    for(int i = 0; i < rows; i++){
        // نضيفو للمصفوفة الاحادية العنصر الحالي من الصف الأخير
        // results[tracker] = array[columns - 1][i];
        *(result + tracker) = (*(array + columns - 1) + i );
        //نزيدو طولها
        tracker++;
    }
    // نطبعو المصفوفة الثنائية
    print2DArray(array, rows , columns);
    cout << endl;
    // نطبعو المصفوفة الجديدة الاحادية
    print1DArray(result, columns);

}

int main(){

    // لو مش فاهم المؤشرات شوف الملاحظة ف سؤال 3

    int array[100][100] , rows , columns;
    cin >> rows >> columns;

    read2DArray(array , rows , columns);

    Throw(array , rows , columns);

    return 0;

}
```

باستخدام المؤشرات، أكتب دالة (ADD) تستقبل مصفوفتين ثنائيتين وأبعاد كل منهما، ثم تجمع المصفوفتين وتخزن ناتج الجمع في مصفوفة أخرى، ثم تستدعي دالة الطباعة للمصفوفة التي تمثل ناتج الجمع.

```
#include <iostream>

using namespace std;

void print2DArray(int array[][500] , int rows , int columns){

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++){
            cout <<*( *(array+i)+j) << " ";

            cout << endl;

        }
    }

}

void Add(int arrayA[][100], int aRows , int aCols , int arrayB[][100], int bRows ,
int bCols){
    int result[500][500];
    // نفترضو اكبر صف و عمود هم متاع اول مصفوفة
    int biggestRows =aRows , biggestCols =aCols;
    // لو طول اعمدة او صفوف b اكبر يخليهم هم الاكبر
    if(aRows < bRows)
        biggestRows = bRows;
    if(aCols < bCols)
        biggestCols = bCols;

    // بنديرو حلقتنا ع اساس الاكبر
    for(int i = 0; i < biggestRows; i++){
        for(int j = 0; j < biggestCols; j++){
            int sum = 0;
            // نتأكدو اننا مطلعناش من طول المصفوفة a ف حالة كانت هي اصغر
            if(i < aRows && j < aCols)
                sum += *( *(arrayA+i)+j) ;
            // نتأكدو اننا مطلعناش من طول المصفوفة b ف حالة كانت هي اصغر
            if(i < bRows && j < bCols)
                sum += *( *(arrayB+i)+j);
            // نخزنو نتيجة جمعهم فالمصفوفة الجديدة
            *( *(result+i)+j) = sum;

        }
    }
    // نطبعو مصفوفة المجموع
    print2DArray(result , biggestRows , biggestCols);
}

void read2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++){
            cin >> *(*(array + i) + j);

        }
    }

}

int main(){
    // لو مش فاهم المؤشرات شوف الملاحظة ف سؤال 3

    int aArray[100][100] , bArray[100][100] , aRows,aCols,bRows,bCols;

    cin >> aRows >> aCols >> bRows >> bCols;
    // نطبعو كل مصفوفة ب ابعادها
    read2DArray(aArray, aRows, aCols);
    read2DArray(bArray, bRows, bCols);

    Add(aArray, aRows, aCols, bArray, bRows, bCols);

    return 0;
}
```


6. أكتب برنامجاً يقرأ عدداً من أسماء المواد (أسماء غير متصلة)، ثم يضيف رمز كل مادة بعدها، مع الحفاظ على مسافة بين الاسم والرمز، ثم يطبع الأسماء برموزها.
ملاحظة: يمكنك استخدام البالة الجاهزة لحساب طول السلسلة، أما غير ذلك من الدوال الجاهزة فلا يسمح باستخدامها.

```
#include <iostream>
#include <string.h> // بيش ننادو ال
using namespace std;

void readNames(char names[][100], int n){
    // نحولو اي مسافة فالكونسل بيش ميحسبهاش ادخال
    cin.ignore();
    // ناخذو الاسماء
    for(int i = 0 ; i < n ; i++){
        cin.getline(names[i] , 100);
    }
}

void mergeCodes(char name[100], char code[100]){
    // نجيبو طول الاسم و الكود
    int n = strlen(name);
    int m = strlen(code);
    name[n] = ' ';
    for (int i = 0 ; i < m ; i++){
        // نخطو الكود قنهایة الاسم
        name[i+n+1] = code[i];
        // نحددو نهاية الاسم الجديدة
        name[m + n + 1] = '\0';
    }
}

void readCodes(char names[][100], int n ){
    for(int i = 0 ; i < n ; i++){
        // متغير مؤقت للكود
        char code[50];
        cin.getline(code , 100);
        // ندمجو الاسم و الكود
        mergeCodes(names[i], code);
    }
}

void printNames(char names[][100], int n){
    for(int i = 0 ; i < n ; i++){
        cout << names[i] << endl;
    }
}

int main(){
    char names[100][100] , codes[100][100];
    int n;

    cin >> n;
    // ناخذو الاسماء من المستخدم
    readNames(names, n);
    // بعدها الاكواد متاع المواد ويدمجهم مع اسم المادة
    readCodes(names, n);
    // نطبعو النتيجة
    printNames(names, n);

    return 0;
}
```

1. أكتب برنامجاً يقوم بقراءة ومطابقة مصفوفة ثنائية C، عناصرها أعداد صحيحة، ثم يقوم بالآتي:

- بحسب المصفوفة D ثم طبعتها، حيث أن: الأعمدة الزوجية في D تساوي الأعمدة الزوجية في C، والأعمدة الفردية في D تساوي 1.
- بحسب المصفوفة E ثم طبعتها، حيث أن: $E = C + 4 + D$.
- رتب صفوف المصفوفة C تصاعدياً ثم طبع المصفوفة.

```
#include <iostream>

using namespace std;

void read2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++)
        for(int j = 0; j < columns; j++)
            cin >> array[i][j];
}

void print2DArray(int array[][100] , int rows , int columns){

    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++)
            cout << array[i][j] << " ";

        cout << endl;
    }
}

void calculateD(int C[][100] , int D[][100], int rows , int columns){
    for(int i = 0; i < rows; i++){
        for(int j = 0; j < columns; j++){
            if(j % 2 == 0) // لو العمود زوجي
                D[i][j] = C[i][j];
            else // مش واضحة الصورة اظن انها 4
                D[i][j] = 4;
        }
    }
}

void sort2DArrayRows(int array[][100], int rows , int columns){
    int temp;
    for(int i = 0; i < rows; i++)
        for(int j = 0; j < columns - 1; j++)
            // نفس ترتيب اي مصفوفة احادية ودرناها فأسئلة النصف
            for (int s = 0; s < columns - 1; s++)
                if (array[i][s] > array[i][s + 1])
                {
                    temp = array[i][s];
                    array[i][s] = array[i][s + 1];
                    array[i][s + 1] = temp;
                }
}

void calculateE(int C[][100] , int D[][100] , int E[][100], int rows , int columns){
    for(int i = 0; i < rows ; i++)
        for(int j = 0; j < columns; j++)
            // نجمعوهم ببساطة سؤال هدية
            E[i][j] = C[i][j] + D[i][j] + 4;
}

int main(){
    int cRows , cCols;
    int C[100][100] , D[100][100] , E[100][100];

    cin >> cRows >> cCols;
    read2DArray(C , cRows , cCols);
    //أ
    calculateD(C , D , cRows , cCols);
    print2DArray(D , cRows , cCols);
    //ب
    calculateE(C , D , E , cRows , cCols);
    print2DArray(E , cRows , cCols);
    //ج
    sort2DArrayRows(C , cRows , cCols);
    print2DArray(C , cRows , cCols);

    return 0;
}
```