

## اسئلة امتحانات النصفى سابقه للأستاذ فرج الشركسى

فى اسئلة متكررة فى الملف وفى اسئلة فىها  
اختلاف بسيط لو شفت سؤال متكرر ركز فىه مرات  
فىه اختلاف بسيط

### أسئلة المقالى

لو لاحظتو اى خطأ قولولنا

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

السؤال الثاني: أجب عن الآتي:

1. باستخدام النوال الفرعية، أكتب برنامجاً لقراءة وطباعة مصفوفة ثنائية  $B$  حجمها  $n \times n$ ، عناصرها أعداد صحيحة، ثم يقوم بالآتي:
    - أ. يبدل أحد أعمدة المصفوفة  $B$  بعمود آخر، ثم يطبع المصفوفة (أي: يبدل العمود  $x$  بالعمود  $y$ ).
    - ب. يرتب أعمدة المصفوفة  $B$  تنازلياً (أي: يرتب كل صف ترتيباً تنازلياً)، ثم يطبع المصفوفة.
    - ج. يستخرج ويطبع أصغر عدد في المصفوفة  $B$ .
    - د. يقرأ المصفوفة  $X$ ، حيث أن حجمها مساوي لحجم المصفوفة  $B$ ، ثم يحسب ويطبع المصفوفة  $E$ ، حيث أن:  $E = B + X$ .
- \*ملاحظة: الدالة تقوم بوظيفة واحدة فقط.

```
#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 swabColumns(int B[][100], int rows , int columns ){
    int x ,y;
    // x و y نأخذو قيمة
    // يلي هما مواقع الأعمدة يلي سنبدا ببنائهم
    cin >> x >> y ;
    // نقصو من المدخلات 1 بيش لو دخل المستخدم واحد فالمصفوفة يكون 0
    // لان اول عنصر فالمصفوفة 0
    x -= 1;
    y -= 1;

    // هنا بنلفو ع الصفوف
    for(int i = 0 ; i < rows ; i++)
        // وهنا ع كل عمود فالصف الواحد
        for(int j = 0 ; j < columns ; j++)
            // لو العنصر الحالي جزء من العمود يلي نيو نبدلوه يعني عموده يساوي اكس
            if(j == x) {
                // عملية تبديل متغيرين طبيعية
                int temp = B[i][x];
                B[i][x] = B[i][y];
                B[i][y] = temp;
            }
}

void sort2DArray(int B[][100], int rows , int columns ){

    // هنا بنلفو ع الصفوف
    for(int i = 0 ; i < rows ; i++)
        // وهنا ع كل عمود فالصف الواحد اي كل عنصر فالصف
        // بما اننا بنغيرو العنصر الحالي ب يلي قدامها فحالة كان الحالي اصغر
        // فدرنا الفور الاولى توقف قبل اخر عنصر ب 1
        // لان لو مدرناهاش اخر عنصر بيتقارن مع عنصر غير معروف مش جزء من المصفوفة
        for(int j = 0 ; j < columns - 1 ; j++)
            // هنا بنقارنو العنصر الحالي يلي هو جي بكل العناصر الباقية فالصف هذا
            for (int s = 0; s < columns - 1; s++)
                // درنا لو العنصر الحالي اصغر من يلي بعده لان قال ترتيب تنازلي مش تصاعدي
                if (B[i][s] < B[i][s + 1])
                {
                    // عملية تبديل طبيعية
                    int temp = B[i][s];
                    B[i][s] = B[i][s + 1];
                    B[i][s + 1] = temp;
                }
}

int getSmallestNumberIn2DArray(int B[][100], int rows , int columns ){
    // نفترضو اول عنصر هو اصغر عنصر
    int smallest = B[0][0];
    // هنا بنلفو ع الصفوف
    for(int i = 0 ; i < rows ; i++)
        // وهنا ع كل عمود فالصف الواحد
        for(int j = 0 ; j < columns ; j++)
            // لو العنصر الحالي اكبر من اخر اصغر قيمة نخلوه هو اصغر قيمة
            if(B[i][j] > smallest)
                smallest = B[i][j];

    return smallest;
}

// خدينا طول الصفوف بس لان هي مصفوفة مربعة يعني عدد الصفوف هو عدد الأعمدة
void calcE(int B[][100] , int X[][100] , int E[][100] , int rows){

    // هنا بنلفو ع الصفوف
    for(int i = 0 ; i < rows ; i++)
        // وهنا ع كل عمود فالصف الواحد
        for(int j = 0 ; j < rows ; j++)
            E[i][j] = B[i][j] + X[i][j];
}

int main(){

    int n ;
    int B[100][100] , X[100][100] , E[100][100];
    // نأخذو ابعاد المصفوفة
    cin >> n;
    // نعتبر للدالة المصفوفة و عدد اعمدتها و عدد صفوفها
    read2DArray(B , n , n);
    // طباعة المصفوفة
    print2DArray(B,n,n);
    // المطلوب الأول
    swabColumns(B , n , n );
    print2DArray(B,n,n);
    // المطلوب الثاني
    sort2DArray(B , n , n );
    print2DArray(B,n,n);
    // المطلوب الثالث
    int smallestNumber = getSmallestNumberIn2DArray(B , n , n);
    cout << smallestNumber << endl;
    // المطلوب الرابع
    // نأخذو عناصر X
    read2DArray(X , n , n);
    // حساب E
    calcE(B , X , E, n);
    // طباعة E
    print2DArray(E,n,n);

    return 0;
}
```

$$SUM = \frac{(x-1)}{(x-1)!} + \frac{(x-2)^3}{(x-2)!} + \frac{(x-3)^4}{(x-3)!} + \frac{(x-4)^5}{(x-4)!} + \frac{(x-5)^6}{(x-6)!} + \dots + \frac{(x-n)^m}{(x-n)!}$$

```
#include <iostream>

using namespace std;
float fect(float x ){
    // بنخزنو الناتج التراكمي هنا
    float mul = 1;
    for(float i = x ; i > 1 ; i--){
        // لو اكس 5 بينضرب ف 5 بتولي مول 25 بعدين 25 ف 4 وهكذا لتوصل ال الأي 1 بنطلعو
        mul *= i;
    }
    return mul;
}
//بناخدو قيمة اكس و الترتيب متاع المعادلة الحالية
//ونطلعو منه البسط و المقام لانهم نفس المعادلة (بدون تربيع او مضروب)
float genFormula(float x , float index){
    //index = 0 => x-1
    //index = 1 => x-2 ...
    return x - index + 1;
}
float genPower(float index){
    //index = 0 => 1
    //index = 1 => 3 ...
    if(index == 0) return 1;
    return index + 2 ;
}
float calcPower(float x , int power){
    float temp = 1; // بيش الأس كان صفر وملفتش الحلقة يخلي القيمة 1
    for(float po = 0 ; po < power ; po++){ // لو الأس صفر الحلقة هذي مش حتخدم
        temp *= x; // نضربو القيمة ف أكس
    }
    return temp;
}
float getSum(float x , float n ){
    float sum = 0 ;
    //الآن محسوبة معانا هذا ليش درنا او يساوي
    //ايه في خطأ مطبعي فصورة السؤال البسط و المقام إن
    for(int i = 0 ; i <= n ;i++){
        // نجيبو المعادلة
        float founmula = genFormula(x , i);
        //نجيبو كم قيمة الأس
        float power = genPower( i);
        //قيمة البسط بعد رفع المعادلة للأس يلي حصلناه
        float first = calcPower(founmula,power);
        //نجيبو المقام بأننا نجيبو مضروب المعادلة
        float second = fect(founmula);
        //نزيدو ناتج قسمتهم ع المجموع
        sum += (first / second);
    }
    //نرجعو المجموع للماين
    return sum;
}
int main(){

    float x ,n ,sum ;
    cin >> x >> n;
    //نجيبو قيمة المتسلسلة
    sum = getSum(x , n);
    cout << sum;
    return 0;
}
```

3. باستخدام النوال الفرعية، أكتب برنامجاً لقراءة وطباعة اسم مادة (غير متصل)، ثم يحذف حرفاً ما من السلسلة كلها وجده فيها. (ملاحظة: لا يسمح باستخدام دوال جاهزة إلا دالة حساب الطول). مثال: السلسلة المُدخلة: Arabic Language، الحرف المحذوف: a، المخرجات: .rbic Lnguge

```
#include <iostream>
#include <string.h> // بيش ننادو ال strlen

using namespace std;

void removeCharacter( char name[100] , char character ){
    //***** هنا نبيك تركز معايا
    // جينا طول الاسم بدالة الجاهزة المسموحة
    int n = strlen( name );
    // بنلفو ع حروف الأسم
    for (int i = 0; i < n; i++)
        // لو الحرف الحالي مساوي للحرف يلي بنحذفوه
        if( name[i] == character )
        {
            // بنوخرو كل الحروف يلي بعده خطوة بحيث انه الحرف يلي بعده بياخذ مكانه
            for ( int j = i ; j < n - 1 ; j++ )
                name[j] = name[j+1];

            // هنا الجزء المهم

            // هنا قلنا للكومبايلر ان الموقع يلي كان اخر حرف فلأسم ولا نهايتها
            // نهاية النص فشار بتكون \0
            name[n - 1] = '\0';
            // بنعاودو ننادو الدالة بنفس المدخلات لكن بعد نقص الطول 1
            return removeCharacter(name , character );
        }
}

int main(){
    char name[100] , character;

    // خدينا من المستخدم الاسم
    cin.getline(name, 100);
    // طبعنا الاسم زي مقال فسؤال
    cout << name << endl;
    // بنجيبو الحرف يلي المستخدم يبي يحذفه
    cin >> character;
    // بنحذفو الحرف من الاسم مهما كان عدد المرات يلي تكرر فيها
    removeCharacter( name , character );
    // نطبعو النتيجة
    cout << name;

    return 0;
}
```

2. باستخدام النوال الفرعية، أكتب برنامجاً لحساب وطباعة SUM، حيث أن:

علماً أن:  $X=100$ ، مع عدم السماح باستخدام النوال الجاهزة.

$$SUM = X + ((x-1)^2 + y^3) + ((x-2)^4 + y^5) + ((x-3)^6 + y^7) + ((x-4)^8 + y^9) + \dots + (1^n + y^m)$$

```
#include <iostream>

using namespace std;

int genPower(int index){
    //index = 1 => 2
    //index = 2 => 4 ...

    return index * 2 ;
}

int calcPower(int x , int power){
    ببش الأس كان صفر وملفتش الحلقة يخلي القيمة 1
    for(int po = 0 ; po < power ; po++){ //لو الأس صفر الحلقة هذي مش حتخدم
        temp *= x; //نضربو القيمة ف أكس
    }
    return temp;
}

int getSum(int x , int y ,int X ){
    بدينا من ال 100 يلي قال عليها ف أكس
    int sum = X ;
    //بدينا من ال 1 لان مش محتاجين لما تكون 0
    //بنوقفو قبل ما ال الأي تساوي اكس لان فالتستسلة مجموع طرحها من الأكس فالاخير لازم يساوي 1
    for(int i = 1 ; i < x ;i++){
        int first = x - i ;
        //نجيبو كم قيمة الأس
        int power = genPower( i);
        //قيمة أول حد من العملية
        first = calcPower(first,power);
        //قيمة ثاني حد مرفوع لأس اعلى من اس اول حد بواحد
        int second = calcPower(y ,power+1);
        //نزيدو ناتج جمعهم ع المجموع
        sum += (first + second);
    }
    //نرجعو المجموع للماين
    return sum;
}

int main(){
    //حساسة لحالة لحروف C++
    //X != x
    int sum , X= 100 , x ,y ;
    cin >> x >> y ;

    sum = getSum(x,y,X);
    cout << sum ;

    return 0;
}
```



```

● ● ●

#include <iostream>
#include <string.h> // ببش ننادو ال

using namespace std;

int findSpaceIndex(char name[100]){
    // طول الاسم
    int n = strlen(name);
    // لشرط بس
    int i = 0;
    while (i < n){
        // لو لقي المسافة بطلع من الدالة ويرجع موقعها
        if (name[i] == ' ')
            return i;
        i++;
    }
    // لو تم الاسم وملقاهاش يرجع -1 لان لازم نرجعو شي
    return -1;
}

void copy( char A[100] , char B[100]){
    // الطول الجديد
    int n = strlen(A);
    // نقل عناصر زي اي مصفوفة
    for(int i = 0 ; i < n ; i++){

        B[i] = A[i];
    }
    // نحددو ان هذي نهاية المتغير المتسوخ الجديدة
    B[n] = '\0';
}

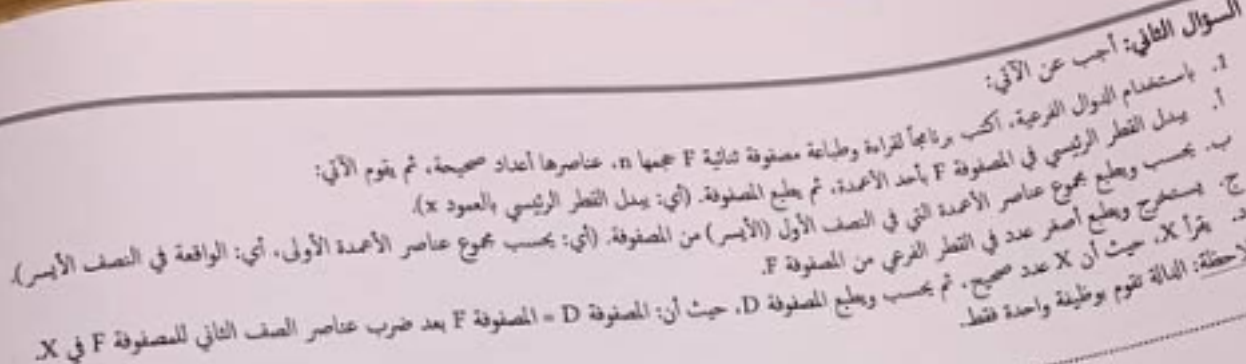
void appendFatherName(char name[100] , char father[100]){
    // بنجمعو الاسم هنا مؤقتا
    char fullName[100];
    // نعرفو طول الاسم و اللقب بدالة المسموحة
    int n = strlen(name);
    // طول اسم الأب
    int fatherNameLength = strlen(father);
    // ببش نعرفو طول الاسم بروحه و طول اللقب بروحه
    // الدالة هذي بتلقا المسافة يلي بيناتهم وتعطينا موقعها/
    // يلي قبلها الاسم ويلى بعدها هو اللقب/
    int spaceIndex = findSpaceIndex(name);
    // ببش نعرفو وين بيتم اسم الأب جمعنا طول اسم الأب مع موقع المسافة
    // ملاحظة ان موقع المسافة بعد الاسم الاول ب 1
    // يعني هكي حسبنا صح لان الطول لازم يكون زايد ب 1
    int fatherEndIndex = fatherNameLength + spaceIndex ;

    // نبدو تكوين الاسم الجديد
    // هذا الاسم الأول بنوقفو زي مقلنا قبل موقع المسافة نقدرو نعتبروه N في المصفوفات/
    for(int i = 0 ; i < spaceIndex ; i++){
        fullName[i] = name[i];
    }
    // توا مكان المسافة الاولى بنخطو مسافة ببش متلصقش الدنيا
    fullName[spaceIndex] = ' ';
    // توا بنضيفو اسم الأب
    // بنبدو من اول موقع بعد المسافة يلي ضفناها/
    // ونوقفو فنهاية اسم الأب لاحظ ان اصغر من او يساوي لان نبو كل حرف
    for(int i = spaceIndex + 1 ; i <= fatherEndIndex; i++){
        // هتا ببش تبدو من بداية اسم الأب لغينا قيمة i المبدئية
        // father[0] , father[1] ... يعني بيكون هكي/
        fullName[i] = father[i - spaceIndex - 1];
    }
    // توا بعد الاسم متاع الاب بنزيدو مسافة
    fullName[fatherEndIndex + 1] = ' ';
    // نحددو موقع نهاية المصفوفة يعني بالعربي طولها الجديد (n)
    // الزائد 1 هذي بسبب اننا زدنا مسافة بين اسم الأب و الاسم الأول
    int fullNameEnd = n + fatherNameLength + 1;
    // بنبدو نضيفو اللقب وبنبدو من بعد اخر مسافة ضفناها
    // ونوقفو قبل الطول الجديد يعني كأنها n
    for(int i = fatherEndIndex + 2 ; i < fullNameEnd ; i++){
        // نبو نرجعو للقب فالمتغير الاول/
        // فنقصنا يلي زدناه ويلى هو 1 متاع المسافة و طول اسم الأب/
        fullName[i] = name[i - fatherNameLength - 1];
    }
    // نقولو للكومبايلر ان هذي نهاية الاسم/
    fullName[fullNameEnd] = '\0';
    // توا ننسخو الاسم الكامل للإسم الأول لان الإسم الكامل مؤقت بس
    // اعلنا عليه هنا ف بينحذف اول متتم الدالة هذي
    copy(fullName , name);
}

int main(){
    char name[100];
    char father[100];

    // ناخذو الاسم و اللقب
    cin.getline(name, 100);
    // طبعناهم زي مطلب فسؤال/
    cout << name << endl;
    // ناخذو اسم الأب/
    cin.get(father, 100);
    // نضيفو اسم الأب بين اللقب و الاسم
    appendFatherName(name , father );
    // نطبعو الإسم الكامل
    cout << name << endl;
    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++)
            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 swabMainD( int F[][100], int n){
    int x ;
    cin >> x;
    //0 بنقصو من المدخلات 1 ببش لو دخل المتستخدم واحد فالمصفوفة يكون
    //0 لان اول عمود فالمصفوفة
    x -= 1;
    // هنا بنلقو ع الصفوف
    for(int i = 0; i < n; i++)
        //وهنا ع كل عمود فالصف الواحد
        for(int j = 0; j < n; j++)
            // لو الصف يساوي العمود معناها حني فالقطر الرئيسي
            if(j == i ){
                // عملية تبديل بين العنصر يلي فالصف الحالي فالقطر الرئيسي
                // و العنصر فالصف الحالي فالعمود المطلوب
                int temp = F[i][x];
                F[i][x] = F[i][j] ;
                F[i][j] = temp;
            }
}

int sumHalfArray( int F[][100], int n){

    int sum = 0;

    for(int i = 0; i < n; i++)
        // هكي بنوقفو فنص المصفوفة يعني العناصر يلي ع ليسار بس
        for(int j = 0; j < (n / 2); j++)
            // نضيفو العنصر للمجموع

            sum += F[i][j];

    return sum;
}

int findSmallestNumberInSubD ( int F[][100] , int n){
    //نفترضو ان اول قيمة فالقطر الفرعي هي اصغر قيمة//
    int smallest = F[0 ][n - 1];

    for(int i = 0 ; i < n ; i++)
        for(int j = 0 ; j < n ; j++)
            // القطر الفرعي هو لما مجموع الصف زائد العمود يساوي طول المصفوفة زائد 1
            // لكن بما اننا فبرمجة مش رياضة المصفوفة عندنا تبدا من 0 مش 1
            // ف ببش نوازنو المعادلة جمعنا الصف مع العمود و تحققنا هل يساوي الطول ناقص 1
            // مثال
            // في الرياضة 1+3 = 4 في حالة المصفوفة طولها 3
            // في البرمجة 0+2 = 2 في حالة المصفوفة طولها 3

            //ولو تحقق الشرط نشوفو هل اصغر من اخر اصغر قيمة عندنا//
            if((i+j) == n-1 && F[i][j] < smallest)
                smallest = F[i][j];

    return smallest;
}

void calcD (int F[][100] , int D[][100] , int n){
    int x;
    // ناخذو قيمة x
    cin >> x;

    for(int i = 0; i < n; i++)
        for(int j = 0; j < n; j++)
            // الصف الثاني نضربو القيمة ف أكس
            if(i == 1)
                D[i][j] = F[i][j] * x;
            else
                // لو اي صف ثاني نخطوها كيف ماهايا
                D[i][j] = F[i][j];
}

int main(){

    int n ;
    int F[100][100] , D[100][100];
    // ناخذو ابعاد المصفوفة
    cin >> n;
    // نبعثو للدالة المصفوفة و عدد اعمدتها و عدد صفوفها
    read2DArray(F , n , n);
    // طباعة المصفوفة
    print2DArray(F,n,n);
    // المطلوب الأول
    // تبديل عمود بالقطر الرئيسي
    swabMainD(F , n);
    print2DArray(F,n,n);
    // المطلوب الثاني
    int sum = sumHalfArray(F , n);
    cout << sum;
    // المطلوب الثالث
    // : واخذوا راحتى فإسماء شوية مش مهم
    // هذا بيلقى اصغر قيمة فالقطر الفرعي
    int smallestValue = findSmallestNumberInSubD(F , n);
    cout << smallestValue;
    // المطلوب الرابع
    calcD(F ,D, n);
    print2DArray(D , n , n);

    return 0;
}
```