ملاحظات عامة:

* تتيح لغة Java إمكانية دمج عمليات التصريح والحجز والإسناد للجزة بتعليمة واحدة كما في المثال التالي:

- `int A[] = {1,2,3,4,5};`

هنا تم: التصريح عن المجزة العددية A.
و حجز 5 أماكن في الذاكرة لعناصرها الخمسة.
و الإسناد الفوري لتم هذه العناصر بالترتيب.

- نستخدم هذه الطريقة إذا كانت قيم عناصر المجزة معروفة لدينا مسبقاً.
- ويمكن أن يتم ذلك أيضاً بالنسبة للمصفوفات كما في المثال التالي:

- `int A[][] = {{1,2},{3,4},{5,6}};` $\Leftrightarrow A=\begin{bmatrix}1 & 2\\ 3 & 4\\ 5 & 6\end{bmatrix}$

هنا تم: التصريح عن المصفوفة العددية A.
و تحديد عدد أسطرها 3 وعدد أعمدتها 2 وحجز أماكن في الذاكرة لهذه العناصر.
و الإسناد الفوري لتم هذه العناصر

* يمكن إجراء عملية إسناد مجزة إلى مجزة أخرى فوراً حتى قبل حجز عناصر الثانية كما في المثال التالي:

- `int A[] = {7, 11, 14, 19, 20}`
  `int B[] = A;` $\Leftrightarrow$
  ```
  int B[];
  B = A;
  ```

ولكن في هذه الحالة يصبح A,B اسمين لمكان واحد في الذاكرة.
أي إجراء أي تغيير على أحدهما يعني إجراءه على الآخر.
أما إذا أردنا فصل عنوانيهما في الذاكرة وتعديل كل واحدة على حدة فإننا نكتب:

```
int A[] = {7, 11, 14, 19, 20}
int B[] = new int[5];
for (int i=0; i<A.Length; i++)
    B[i] = A[i];
```


Scanned by CamScanner

تمرين: اكتب برنامجاً بلغة الجافا يقوم بإدخال مصفوفة محارف من البعد mxn حيث n , m عددان طبيعيان مدخلان أكبر من 1 وأصغر من 100، ثم ينجز التالي:

1. إيجاد وطباعة أكبر عنصر في المصفوفة المدخلة مع دليليه.
2. طباعة الكلمة المشكلة من عناصر القطر الرئيسي
3. طباعة معكوس الكلمة المشكلة من عناصر القطر الثانوي.
4. طباعة المصفوفة المدخلة على شكل مصفوفة.
5. طباعة المصفوفة على شكل مصفوفة بعد حذف عناصر سطرها الأول وعمودها الأخير.

الحل:

```
class Mymatrix{
public static void main (String args [])
{ char c[][];     int n,m;
  do { n=Stdin.readInt(); } while (n<=1 || n>100);
  do {m=Stdin.readInt(); } while (m<=1 || m>100);
  c = new char [n][m];
  for (int i=0 ; i<c.length ; i++)
     for( int j=0 ; j<c[i].length ; j++)
        c[i][j] = Stdin.readChar();
```

حل الطلب الأول:

```
char max = c[0][0] ;
int k=0, L=0;
for (int i=0 ; i<c.length ; i++)
   for (int j=0 ; j<c[i].length ; j++)
      if (c[i][j] >max) { max=c[i][j];
                          k = i;
                          L = j; }
System.out.println ("max=" + max + "," + k + "," + L);
```

هنا يطبع قيمة ودليلي أول قيمة صادفها لأكبر عنصر في المصفوفة

Scanned by CamScanner

حل الطلبين الثاني والثالث :

```
if (n!=m)   System.out.println(" error ::c is not square");
else { for (int i=0 ; i<c.length ; i++)
          System.out.print (c[i][i]);

       for (k = c.Length -1 ; k>=0 ; k--)
        System.out. print (c[k][n-k-1]);
     }
```

حل الطلب الرابع :

```
for (int i=0 ; i<c.length ; i++)
{   for (int j=0 ; j <c[i]. Length ; j++)
       System.out.print (c[i][j] + "  ");
    System.out.println ();
}
```

حل الطلب الخامس :

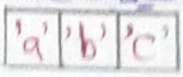
```
for (int i=1 ; i<c.length ; i++)
{  for (int j=0 ; j<c[i].length-1 ; j++)
       System.out.print (c[i][j] + "  ");
    System.out.println ();
} // end for

} // end main
} // end class
```

ALAUIB.net
أبناء الحضارة
Scanned by CamScanner

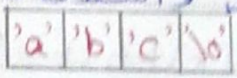
* ما هو الفرق بين متجهة المحارف، ومتحول مركب من النوع String ؟

لنفرض أننا نريد تخزين السلسلة الحرفية abc.

* لو قمنا بتعريف متجهة محارف char A[] مؤلفة من ثلاث عناصر، وأسندنا للأول القيمة 'a' وللثاني القيمة 'b' وللثالث القيمة 'c' فيكون الحجز في الذاكرة على الشكل التالي:



* أما لو قمنا بتعريف متحول من النوع String وأسندنا له القيمة abc فيكون الحجز في الذاكرة على الشكل التالي:



حيث '\0' هي نهاية السلسلة الحرفية.

---

* إن النوع String هو عبارة عن صنف في لغة الجافا، وهي تحتوي على عدد كبير من الدوال الأعضاء مثل:

1) char charAt (int index)

يعيد المحرف المحدد بالدليل index.

2) int indexOf (char ch)

ترجع دليل أول ظهور للمحرف ch في السلسلة.

3) int length ()

ترجع طول السلسلة.

4) int compareTo (String s)

تقارن طولي السلسلتين:

- 0 ⟹ تطابق في عدد المحارف
- <0 ⟹ s أطول.
- >0 ⟹ s أقصر.

5) char [] toCharArray()

ترجع محارف السلسلة إلى متجهة محارف.

6) String toLowerCase()

تحول أحرف السلسلة إلى أحرف صغيرة.

7) String toUpperCase()

تحول أحرف السلسلة إلى أحرف كبيرة.

ـــ وسندرس مفهوم الصفوف ودوالها بالتفصيل ابتداءً من المحاضرة القادمة.

Scanned by CamScanner