

◀ دكتور الملاءة: سير جعفر

◀ عنوان المحاضرة: تمرين: صفوة حروف



- أكتب برنامجاً يقوم بإدخال صفوة حروف من البعد  $n \times m$  ، يصيغ  $n, m$   
 أعداد طريفة موزعة أكبر من الواحد ثم يبرز ما يلي:
1. طباعة الكلمة المشكّلة من عناصر قطرها الرئيسي.
  2. طباعة الكلمة المشكّلة من عناصر قطرها الثانوي.
  3. البحث عن الحرف المدخل  $ch$  فيما إذا كان موجوداً في الصفوة أم لا، مع تحديد مكانه في حال وجوده، (اخذ يد سطر وعمود).
  4. طباعة مقول الصفوة على شكل صفوة.
  5. طباعة الصفوة على شكل صفوة بعد حذف عناصر سطرها الثاني وعمودها ما قبل الأخير.

الحل:

```
class MyMatrix {
```

```
public static void main (String args[]) {
```

```
char C[][];
```

المطلوب صفوة حروف لذلك نرّحنا ←

عن صفوة حروف وأسمائها C

```
int n, m;
```

المصفوفة من البعد  $n \times m$  ، لذلك نرّحنا عن متغيرين صحيحين ←

```
do { n = Stdin.readInt(); } while (n <= 1);
```

```
do { m = Stdin.readInt(); } while (m <= 1);
```

نريد البعدين  $n, m$  أكبر وأما عن الواحد ، لذلك استخدمنا الحلقة `do... while` لتعدد إدخال البعد طالما أنه أصغر أو يساوي الواحد.

c = new char [n][m];

قنا بحجم مكان في الذاكرة المصنوفة ←

```
for (int i=0 ; i < c.length ; i++)
for (int j=0 ; j < c[i].length ; j++)
    c[i][j] = stdin.readChar();
```

حلقة for للأسطر  
والأعمدة لإدخال  
عناصر المصفوفة.

```
if (c.length != c[0].length) System.out.print("error");①
else { for (int i=0 ; i < c.length ; i++)②
        System.out.print (c[i][i]);
```

```
for (int i=0 ; i < c.length ; i++)③
    System.out.print (c[i][c.length - 1 - i]);
```

① حلقة شريطة - إذا كان عدد الأسطر لا يساوي عدد الأعمدة تعطي خطأ  
② وإلا تقوم بطباعة حارف القطر الرئيسي.  
« استخدمنا التعليمة if هنا لأننا نريد المصفوفة مربعة »  
③ طباعة حارف القطر الثانوي.

char ch = stdin.readChar();<sup>④</sup>

boolean t = false;<sup>⑤</sup>

int a, b;

```
for (int i=0 ; i < c.length ; i++)⑥
```

```
for (int j=0 ; j < c[i].length ; j++)
```

```
    if (c[i][j] == ch) { t = true;⑦
```

a = i;

b = j; }

if (t == true) S.O.p("exist" + a + " " + b); } ⑧  
else S.O.p("not exist");

④ مرتجنا عن متغير من النوع الممر . وأدخلنا قيمته (وزيد العبد عنه في المصفوفة)

⑤ عرفنا متغير بولياي (t) وأعطيناه القيمة (false) حينئذياً .

⑥ طبقنا for للترور على جميع عناصر المصفوفة .

⑦ في طب وجود قيمة العنصر الممر في المصفوفة يجعل قيمة المتحول البولياي (true)

ويسند قيمة السطر إلى a والعنود إلى b .

⑧ نطبع النتيجة مع ذلك السطر والعنود ، وإلا فإنه غير موجود ونطبع النتيجة .

```
char C1 = new char [C[0].length][C.length];
for(int i=0 ; i < C.length ; i++)
for(int j=0 ; j < C[0].length ; j++)
    C1[j][i] = C[i][j]; } ⑨
```

```
for(int i=0 ; i < C1.length ; i++) {
for(int j=0 ; j < C1[0].length ; j++)
    S.O.p(C1[i][j] + " ");
    S.O.p("\n"); } } ⑩
```

سنقول مصفوفة يتكون قلب أسطرها أعده وأعدادها أسطر .

⑨ طبقنا for للترور على جميع العناصر ومن ثم قلب الأسطر أعده والأعده أسطر .

⑩ طباعة المنقول بشكل مصفوفة : يطبع عناصر كل سطر ثم ينزل سطرأ في كل دورة .

```
for (int i=0 ; i < c.length ; i++) {
    for (int j=0 ; j < c[0].length ; j++)
        if (i != 1 && j != c[0].length - 2)
            S.O.P (c[i][j] + " ");
        else S.O.P (" " + " ");
    S.O.Pln ();
}
```

المرور على جميع العناصر

وإذا وصلنا إلى ذلك السطر التالي أو دليل العود قبل الأخير نقوم بطباعة فراغ.

(قوس نهاية، لصق) (قوس نهاية main)

ملاحظات:

يمكن كتابة if (t) بدلاً من if (t == true) و if (!t) بدلاً من if (t == false)

كتابة S.O.P اختصاراً للطباعة System.out.print (لكن عند كتابة البرنامج على الحاسوب لا يمكن اختصارها)

c.length تدل على عدد أسطر المصفوفة.

c[i].length (أو c[0].length) تدل على عدد عناصر المجموعة المتكعبة بالمضمار السطر (i) (أو السطر (0)) ، وبالتالي تدل على عدد أعمدة المصفوفة.

في الخطوة ① : لا يمكننا كتابة c[i].length بدلاً من c[0].length في كل هذه الحالة ، لأننا كنا قد عرفنا المتغير (i) كمعادلة للخطوة for ضمن الحلقة (أي أنه غير معرف خارج الحلقة) ، لذلك بعد خروجه من الحلقة لا يتعرف ال Compiler على (i) في حال استخدامه.