



## lesson 62 (Find a number)

إذا كان لدينا **array** وبه مجموعة من الأرقام ونريد عمل بحث بين تلك الأرقام عن رقم معين سيقوم المستخدم بإدخاله

دعنا نكتب البرنامج بطريقة يقوم بها الكثيرون وهي كالتالي :

```
#include <stdio.h>
```

```
int main( ) {  
    int i, y, x[ ] = { 8,5,6,90,3,1 };  
    // يمكننا حجز Array بدون الحاجة لكتابة عددها  
    printf("Enter a number: ");  
    scanf("%d", &y);  
    for (i = 0; i < 6; i++)  
        if (x[i] == y)  
            printf("Found\n");  
        else  
            printf("Not Found\n");  
}
```

( قم بتجربة الكود بنفسك واضغط هنا )

هنا إذا قام المستخدم بإدخال رقم، سيقوم البرنامج بالمرور على كل رقم أثناء عمل ال **for** و سينفذ ما في أمر **if** فإذا كان في هذه الدورة الشرط **false** والرقم غير موجود سيطلع **Not found** وهذا معناه انه إذا لم يجد الرقم في أول دورة أو ثاني دورة أو الثالثة أو الرابعة وهكذا سيطلع **not found**



و على الجانب الآخر، إذا كان الرقم موجود في ال array بشكل متكرر، إذا في كل دورة يكون فيها الشرط **true** سيقوم بطباعة found وهذا معناه انه إذا وجد الرقم في أول دورة و ثانی دورة و الثالثة و الرابعة وهكذا سيطبع found أكثر من مرة و هذا أيضا خطأ

إذا ما المشكلة و كيف يتم حلها ؟

من الخبرة ستكتشف أنه عندما نجد أمر طباعة في البرنامج تم تنفيذه بكثرة و قام بالطباعة بعدد كثير، ونحن كنا بحاجة إلى أمر طباعة واحد يكون أمر الطباعة داخل أمر **for**

**input:**

Enter a number: 1

**output:**

Not Found

Not Found

Not Found

Not Found

Not Found

Found

هناك بعض الأشخاص يقوموا باستخدام **return 0** أو **break** كي يتوقف البرنامج دعنا نقوم باستخدامها

```
#include <stdio.h>
```

```
int main( ) {
```

```
    int i, y, x[ ] = { 8,5,6,90,3,1 };
```

```
    // يمكننا حيز Array بدون الحاجة لكتابة عددها
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &y);
```



```
for (i = 0; i < 6; i++)
    if (x[i] == y){
        printf("Found\n");
        return 0;
    }
else
    printf("Not Found\n");

}
```

**input:** 8

**output:**

found

( قم بتجربة الكود بنفسك واضغط هنا )

هذا إذا كان الرقم أول ال **Array** ، إما إذا كان في منتصف ال **Array** أو آخرها سيقوم أيضاً بطباعة not found إلى أن يصل إلى الرقم الذي نبحث عنه

**input:** 3

**output:**

Not Found

Not Found

Not Found

Not Found

Found

هنا يوجد امر **for** يقوم بالدوران على كل عنصر إذا لم يجده يطبع not found إلى أن يجده



وبالتالى الحل الصحيح لهذا البرنامج يكون بأننا نقوم بإخراج أمر الطباعة خارج **for** ونقوم بعمل متغير يقوم بالعدد إذا وجده يضيف واحد وسيكون كالتالى :

```
#include <stdio.h>
int main( ) {
    int i, y, x[ ] = { 8,5,6,90,3,1 },found=0;
    // يمكننا حجز Array بدون الحاجة لكتابة عددها
    printf("Enter a number: ");
    scanf("%d", &y);
    for (i = 0; i < 6; i++)
        found += x[i] == y;
    // إذا كان الرقم موجود هذا معناه true معناه واحد//
    // إذا كانت found اكبر من أو تساوى واحد فهذا معناه ان الرقم موجود//

    if (found > 0)
        printf("found");
    else
        printf("Not found");
}
```

input:

1

output:

found



( قم بتجربة الكود بنفسك واضغط هنا )

وهناك طريقة أخرى

```
#include <stdio.h>
```

```
int main( ) {
```

```
    int i, y, x[ ] = { 8,5,6,90,3,1 };
```

```
    // يمكننا حجز Array بدون الحاجة لكتابة عددها
```

```
    printf("Enter a number: ");
```

```
    scanf("%d", &y);
```

```
    for (i = 0; i < 6; i++)
```

```
        if (x[i] == y){
```

```
            printf("Found\n");
```

```
            return 0;
```

```
            // إذا تحقق الشرط يقوم بالطباعة
```

```
            // ويقوم بإغلاق البرنامج
```

```
        }
```

```
    printf("Not Found\n");
```

```
    // إذا لم يتحقق الشرط سيخرج من أمر for
```

```
    // Not found ويطبع
```

```
}
```

أمر `return 0` هو أمر يقوم بإغلاق البرنامج بالكامل ، وبالتالي إذا كان هناك كود نحتاج إليه أن يعمل، بعد تنفيذ `return 0` لن يعمل



---

( قم بتجربة الكود بنفسك واضغط هنا )