



## lesson 57 (Integer)

فى هذا الدرس نريد عمل برنامج يقوم بعدد الخانات ( digits ) الموجودة داخل رقم ما، على سبيل المثال إذا كان لدينا رقم 5896 فهو يحتوى على 4 digits ، إذا كان لدينا 856974 فهذا الرقم يحتوى على 6 digits، ورقم 52 يحتوى على 2 digits وهكذا. لكن ما الفرق بين علامة ( / ) القسمة ، و ( % ) باقى القسمة ؟ عند استخدام علامة القسمة / ، إذا قمنا بعملية قسمة عدد صحيح

مثال: 85/10

سيكون الناتج 8.5 لكن لأن 85 محجوز int

سيكون الناتج 8 لأن هناك علامة عشرية

مثال: 429/10

سيكون الناتج 42.9 لكن لأن 429 محجوز int

سيكون الناتج 42

لذلك هناك علامة % التي درسناها من قبل وهى تعنى باقى القسمة

إذا كان لدينا رقم يساوي 4123

إذا قمنا بعمل باقى قسمة 10%4123 سيكون الناتج 3

إذا قمنا بعمل باقى قسمة 10%412 سيكون الناتج 2

إذا قمنا بعمل باقى قسمة 10%41 سيكون الناتج 1

إذا قمنا بعمل باقى قسمة 10%4 سيكون الناتج 4

وبالتالى فى هذا البرنامج يطلب منا عدد الـ digits فى الرقم الواحد ويكون البرنامج كالتالى :



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

```
int main( ) {
```

```
    int x,r=0;
```

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

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

```
    // قام المستخدم بإدخال الرقم
```

```
    while (x) {
```

```
        // قمنا بعملها لأننا لا نعلم كم عدد الأرقام التي داخل الرقم الواحد
```

```
        x /= 10;
```

```
        // قمنا بالقسمة على 10 كي نخرج رقم من x
```

```
        r++;
```

```
        // نقوم بعدد اللفات لعد الأرقام
```

```
    }
```

```
    printf("%d", r);
```

```
}
```

**input:**

369852741

**output:**

9

( قم بتجربة الكود بنفسك واضغط هنا )

هنا في هذا البرنامج يعمل مرة واحدة فقط ، لكن ماذا إذا أردنا أن نجعله يعمل دائماً هناك طريقتين :

إما أن نضع كل الكود الموجود داخل main في `while (true){ }`



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

```
int main( ) {
    while ( true ){
        int x,r=0;
        printf("Enter number: ");
        scanf("%d", &x);
        while (x) {
            x /= 10;
            r++;
        }
        printf("%d", r);
    }
}
```

الطريقة الثانية :

هناك أمر في لغة **C** يسمى **goto**

ولأستخدامه نضع الكود المراد تنفيذه بين أى كلمه نريد الذهاب إليها متبوعة : بنقطتان فوق بعض

وأمر **goto** متبوع بالكلمة المراد الذهاب إليها .

بمعنى أنه يقول للبرنامج : قم بالرجوع إلى هذه الكلمة ونفذ الأكواد الموجودة بينها

فى البرنامج السابق سنستخدم فيه هذا الأمر

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

```
int main( ) {
    int x,r;
```



start:

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

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

// قام المستخدم بإدخال الرقم

```
r=0;
```

// كى يبدأ العد من الصفر من جديد فى كل دورة

```
while (x) {
```

// قمنا بعملها لأننا لا نعلم كم عدد الأرقام التى داخل الرقم الواحد

```
x /= 10;
```

// قمنا بالقسمة على 10 كى نخرج رقم من x

```
r++;
```

// نقوم بعد عدد اللفات لعد الأرقام

```
}
```

```
printf("%d", r);
```

```
goto start;
```

```
}
```

### ( قم بتجربة الكود بنفسك واضغط هنا )

هنا كل مرة ينفذ الكمبيوتر أمر الطباعة سيجد بعده أمر **goto** ويخبره أن يذهب إلى كلمة **start** يقوم بالرجوع إلى كلمة **start** ثم تنفيذ الأكواد التى تليها

لكن هنا فى هذا البرنامج هناك خطأ يقع فيه البعض وهو تعريف **r=0** فوق **start** وهنا فى كل دورة يقوم البرنامج بجمع القيمة القديمة فى اللفة القديمة مع القيمة الجديدة لكن هنا نحتاج فى كل دورة فى أمر **goto** أن تكون **r** تساوى صفر

هناك ملاحظة أمر **goto** لكى يعمل بصورة صحيحة على أى **compiler** للغة C



لابد أن يكون تحت تعريف المتغيرات

```
#include <stdio.h>
int main( ) {
    start:
    int x,r;
    // هنا سينتج خطأ لأن المتغيرات تم تعريفها بعد start
    printf("Enter number: ");
    scanf("%d", &x);
    r=0;
    while (x) {
        x /= 10;
        r++;
    }
    printf("%d", r);
    goto start;
}
```