



lesson 17 Binary Program

حتى الآن تعلمنا ما هو ال binary و ال deciaml عرفنا كيف يمكننا تحويل أى رقم binary إلى رقم decimal بسهولة عن طريق ضرب كل digit من الرقم ال binary فى رقم 2 مرفوع لأس يساوى ترتيب هذا ال digit (حيث أن الترتيب يبدأ من 0) ثم نقوم بجمع النواتج.
و بهذا نجد أن هناك نمط يمكن تحويله إلى برنامج يقوم الكمبيوتر بتنفيذه و لكن أولاً علينا فهم بعض المفاهيم .

باقى القسمة (%)

عندما درسنا أنواع المتغيرات عرفنا أن هناك متغير من نوع اسمه float هذا النوع يقبل أن يحمل أرقام عشرية (كسور) و ليس فقط أعداد صحيحة.
فإذا كان هناك متغير من نوع float يساوى 15 و نريد أن نطبع نصف قيمته سيكون الناتج 7.5 ، و لكن درسنا نوع آخر من المتغيرات و نستخدمه بكثرة و هو ال int و هو النوع الذى يحمل الأعداد الصحيحة فقط و لا يقبل الكسور .
فإذا كان هناك متغير من نوع int يساوى 15 و نريد أن نطبع نصف قيمته سيكون الناتج 7 لأن ال 5 تحتوى على رقم 7 متكرر مرتين فعند قسمة $15 / 7 = 2$ و يتبقى 1 ، لأن فى الأصل $14 = 2 * 7$ و ليس 15 .

فباقى القسمة هو ما تبقى من الرقم عند قسمته على عدد ما بحيث هذا الباقي يكون أقل من المقسوم عليه .

يمكننا رؤية الأمر بطريقة أخرى، لحساب باقى قسمة متغير من نوع int على أى رقم، إذا كان يقبل القسمة عليه مباشرة سيكون باقى القسمة 0، مثل $14 / 2$ الناتج 7 و باقى القسمة 0، أما إذا كان لا يقبل القسمة عليه نبدأ بإنقاص 1 من الرقم الذى يتم قسمته إلى أن نصل إلى رقم يقبل القسمة مباشرة، ثم نقوم بجمع كل 1 تم حذفه و هذا المجموع هو باقى القسمة. مثال : $15 / 4$ ، فى البداية ال 15 لا تقبل القسمة على 4 ، إذا نبدأ بإنقاص 1 من ال 15 لتصبح 14، ال 14 لا تقبل القسمة على 4 ، إذا نبدأ بإنقاص 1 من ال 14 لتصبح 13 ،



ال 13 لا تقبل القسمة على 4 ، إذا نبدأ بإنقاص 1 من ال 13 لتصبح 12 ، ال 12 تقبل القسمة على 4 مباشرة و يكون باقى القسمة 0 . إذا باقى قسمة 15 على 4 هو 3 .

لاحظ باقى قسمة أى رقم تكون أقل من المقسوم عليه ، بمعنى إذا كان المقسوم عليه 4 فباقى القسمة ممكن أن تكون 3 أو 2 أو 1 أو 0

كنا إذا أردنا حساب قيمة حاصل القسمة نستخدم علامة سلاش (/) Slash

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

```
int main( ) {
    int num = 14;
    printf(" %d \n ", num / 3 );
}
```

النتائج : 4

إذا كيف نحسب باقى القسمة فى الكود ؟
نستخدم علامة ال (%)

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

```
int main( ) {
    int num = 14;
    printf(" %d \n ", num % 3 );
}
```

النتائج : 2



لربما تسأل نفسك الآن، لماذا تعلمنا باقى القسمة و لماذا هذا القسم ضرورى فى فكرة الدرس و هى تحويل الرقم من binary إلى decimal باستخدام الكمبيوتر !
فى الواقع هناك استخدامان شهيران لمعامل باقى القسمة % :

1. معرفة العدد زوجى أم فردى
2. استخدام خانة الأحاد فى أى رقم

1. لمعرفة أى عدد إذا كان زوجى أم فردى، بكل بساطة سيختبر الكود العدد، إذا كان يقبل القسمة على 2 (باقى قسمته على 2 يساوى 0) سيكون عدد زوجى و عكس ذلك سيكون عدد فردى .
ملاحظة (عكس ذلك) ليس له أى احتمال إلا أن يكون 1 فكما قلنا أن باقى القسمة دائمة يكون أقل من المقسوم عليه و بم أن المقسوم عليه 2 فباقى القسمة إما أن يكون 0 أو 1 .

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

```
int main( ) {
    int num ;
    scanf("%d", &num);
    if ( num % 2 == 0 )
        printf(" %d is even \n ", num );
    else
        printf(" %d is odd \n ", num );
}
```

ملحوظة يمكننا فى شرط ال if أن نكتبه بهذا الشكل :

```
if ( num % 2 )
    printf(" %d is odd \n ", num );
```



else

```
printf(" %d is even \n ", num );
```

لأن داخل أقواس ال if يتم ترجمة الشرط إلى true او false ، ففي حالة كتبنا :

```
if ( num % 2 == 0 )
```

هذا معناه أن الشرط سيتحقق لو كان الرقم زوجي لأن باقي القسمة يساوي 0 ، فكأن المكتوب هو :

```
if ( true )
```

أما في حالة كتبنا :

```
if ( num % 2 )
```

هذا معناه أن الشرط سيتحقق لو كان الرقم فردي لأن باقي القسمة يساوي 1 ، فكأن المكتوب هو :

```
if ( 1 )
```

و الذي يحمل نفس معنى

```
if ( true )
```

و لو كان الرقم زوجي أى أن باقي القسمة هو 0 ، فلن يتحقق الشرط ، فكأن المكتوب هو :

```
if ( 0 )
```

و الذي يحمل نفس معنى

```
if ( false )
```

تذكر أن ليس ال 1 فقط هو ما معناه true و لكن أى شئ غير ال 0 معناه true و لكن في هذا المثال ليس هناك خيار آخر غير ال 1 و ال 0 كما ذكرنا



2. لاستخدام خانة الأحاد، كما ذكرنا أن دائما باقى القسمة يكون أقل من المقسوم عليه و خانة الأحاد فى أى رقم من المؤكد أنها أقل من 10 فإذا قمنا بحساب باقى قسمة أى رقم على 10 سيكون الناتج هو خانة أحاد هذا الرقم و ذلك لأن لا يقبل القسمة على 10 إلا رقم فى خانة أحاده 0 و فى هذه الحالة باقى القسمة سيكون 0 أيضا فهكذا استخدمنا خانة الأحاد و فى حالة كان رقم مختلف عن 0 سيظل البرنامج فى إنقاص 1 من الرقم كله إلى أن يصل إلى رقم خانة أحاده تساوى 00 حتى يقبل القسمة على 10 ، و عند تجميع كل 1 تم حذفه ستحصل على نفس العدد الذى كان فى خانة الأحاد .

مثال 10 % 140، ال 140 تقبل القسمة على 10 و الباقي 0 ، و هو نفس خانة الأحاد فى الرقم 140 .

مثال آخر 10 % 156 ، ال 156 لا تقبل القسمة على 10 ، إذا نبدأ بإنقاص 1 من ال 156 لتصبح 155 ، ال 155 لا تقبل القسمة على 10 ، إذا نبدأ بإنقاص 1 من ال 155 لتصبح 154 ، ال 154 لا تقبل القسمة على 10 ، إذا نبدأ بإنقاص 1 من ال 154 لتصبح 153 ، ال 153 لا تقبل القسمة على 10 ، إذا نبدأ بإنقاص 1 من ال 153 لتصبح 152 ، ال 152 لا تقبل القسمة على 10 ، إذا نبدأ بإنقاص 1 من ال 152 لتصبح 151 ، ال 151 لا تقبل القسمة على 10 ، إذا نبدأ بإنقاص 1 من ال 151 لتصبح 150 ، ال 150 تقبل القسمة على 10 ، إذا مجموع ال 1 الذين تم إنقاصهم من 156 للحصول على رقم يقبل القسمة على 10 و هو (150) يساؤوا 6 ، أى أنه نفس خانة أحاد الرقم الأسمى (156)

بالمناسبة كما كنا فى عملية القسمة العادية الخاصة بال int عندما يكون هناك باقى للقسمة يتم التخلص منه كما ذكرنا فى مثال 15 / 2 يساوى 7.5 و لكن تم التخلص من ال 0.5 فأصبح الناتج النهائى 7

هكذا عند القسمة على 10 دائما ما يتحول رقم الأحاد إلى الرقم بعد العلامة العشرية :

$$150 / 10 = 15.0 \quad \text{الناتج النهائى هو } \Leftarrow 15$$

$$159 / 10 = 15.9 \quad \text{الناتج النهائى هو } \Leftarrow 15$$



$155 = 10 / 15.5$ الناتج النهائى هو $\Leftarrow 15$
 $153 = 10 / 15.3$ الناتج النهائى هو $\Leftarrow 15$

الآن هيا نقوم بعمل برنامج يحول أى رقم من نظام ال decimal إلى binary

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

```
int main( ) {
```

```
    int binary, decimal = 0,y=1;
```

```
    // هنا قمنا بحجز متغيرين
```

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

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

```
    // قام المستخدم بإدخال الرقم المكون من 0 و 1
```

```
    while (binary) {
```

```
        decimal += ( binary % 10 ) * y;
```

```
        // إضافة النواتج إلى متغير الرقم العشري الذى سيتم طباعته فى النهاية
```

```
        binary /= 10;
```

```
        // التخلص من رقم رقم
```

```
        y *= 2;
```

```
        // مضاعفات ال 2
```

```
    }
```

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

```
}
```

(قم بتجربة الكود بنفسك واضغط هنا)

كيف تم عمل البرنامج ؟



تم حجز المتغير الذى سيدخل فيه المستخدم الرقم الذى يريد تحويله (binary) و المتغير الذى سيتم تحويل الرقم فيه (decimal) و بدأ بـ 0 لأنه سيمثل المجموع و متغير y الذى يمثل مضاعفات الـ 2 التى يتم الضرب فيها كل مرة.
تعالى نلقى نظرة على الطريقة التى كنا نقوم بالتحويل باستخدامها فى الدرس السابق :

لتحويل رقم 1010 من binary إلى decimal :

256	128	64	32	16	8	4	2	1
-----	-----	----	----	----	---	---	---	---

1 0 1 0

الحل :

$$0*1 + 1*2 + 0*4 + 1*8 = 10$$

الجدول الذى فيه مضاعفات الـ 2 سيمثله المتغير y ، و الرقم الـ binary هو الرقم الذى سيدخله المستخدم و الرقم الـ decimal النهائى هو مجموع حواصل ضرب كل خانة من الرقم الـ binary فى القيمة المقابلة لها من مضاعفات الـ 2 .

سيتم فصل كل digit فى خانة الأحاد من الرقم الـ binary باستخدام معامل باقى القسمة و ضربه فى مضاعف الـ 2 المناسب لترتيبه .

1 0 1 0

فى الدورة الأولى :

أول رقم فى خانة الأحاد و هو الـ 0 سيتم ضربه فى قيمة الـ y الأولى و هى 1 و سيتم إضافة الناتج على المتغير decimal و بم أن الناتج 0 فلن تتغير قيمة الـ decimal ، ثم سيتم فصل خانة الأحاد هذه من الرقم باستخدام عملية القسمة على 10 ليصبح الرقم 101



بعد أن كان 1010، ثم سيتم مضاعفة ال y من 1 لتصبح 2 لأننا سندخل على الدورة الثانية التي نتعامل مع الخانة الثانية في الرقم ال binary.

1 0 1

في الدورة الثانية :

أول رقم في خانة الأحاد الآن (بعد حذف ال 0) هو ال 1 سيتم ضربه في قيمة ال y الحالية و هي 2 و سيتم إضافة الناتج على المتغير decimal و بم أن الناتج 2 فستصبح قيمة ال decimal تساوى 2 ، ثم سيتم فصل خانة الأحاد هذه من الرقم باستخدام عملية القسمة على 10 ليصبح الرقم 10 بعد أن كان 101، ثم سيتم مضاعفة ال y من 2 لتصبح 4 لأننا سندخل على الدورة الثالثة التي نتعامل مع الخانة الثالثة في الرقم ال binary.

1 0

في الدورة الثالثة :

أول رقم في خانة الأحاد الآن (بعد حذف ال 1) هو ال 0 سيتم ضربه في قيمة ال y الحالية و هي 4 و سيتم إضافة الناتج على المتغير decimal و بم أن الناتج 0 فلن تتغير قيمة ال decimal و ستظل تساوى 2 ، ثم سيتم فصل خانة الأحاد هذه من الرقم باستخدام عملية القسمة على 10 ليصبح الرقم 1 بعد أن كان 10، ثم سيتم مضاعفة ال y من 4 لتصبح 8 لأننا سندخل على الدورة الرابعة التي نتعامل مع الخانة الرابعة في الرقم ال binary.

1

في الدورة الرابعة :

الرقم الوحيد في خانة الأحاد الآن (بعد حذف ال 0) هو ال 1 سيتم ضربه في قيمة ال y الحالية و هي 8 و سيتم إضافة الناتج على المتغير decimal و بم أن الناتج 8 فستصبح قيمة ال decimal تساوى $10 = 8 + 2$ ، ثم سيتم فصل خانة الأحاد هذه من الرقم باستخدام عملية القسمة على 10 ليصبح الرقم 0 بعد أن كان 1 (لأن الناتج 0.1)، ثم سيتم مضاعفة ال y من 8 لتصبح 16 .



عند محاولة الدخول فى الدورة الخامسة سنجد أن شرط ال `while` لم يعد متحقق حيث أن رقم ال `binary` وصل إلى 0.