



lesson 65 Common Mistakes

من المحتمل عندما تشاهد فيديوهات البرمجة على الموقع، تشعر أن البرمجة أمر بسيط تفهمها جيداً، لكن عندما تقوم بكتابة برامج بنفسك ستكتشف أن هناك بعض الأشياء حتى الآن لم تقم بفهمها بصورة صحيحة، لأن البرمجة 10% منها عبارة عن تعليم و 90% تمارين، وهى أنك تقوم بعمل برامج بنفسك، مجرد قراءة الدروس لا يجعلك متميز في البرمجة

لا بد من حل أسئلة وتمارين كثيرة

عند قيامك في البداية بحل تمارين ستجد بعض الأخطاء، لكن هذه الأخطاء هى شئ طبيعى يكون في البداية، يحصل لكل المبتدئين في المجال

في هذا الدرس سنتعرف على بعض الأخطاء الشائعة، لكن يجب التنويه على أنه لا بد من التدريب كثيراً وحل برامج كثيرة ومحاولة عمل برامج لم تقم بدراسها
أول خطأ :

هى علامة = وهنا إذا كان $x = y$ فهى معناها أن قيمة الموجود فى y تنتقل إلى x فى الرياضيات كنا ندرس أن $x = y$ هى نفسها $y = x$ لكن فى البرمجة هذا شئ والمثال الثاني شكل آخر
مثال :

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

```
int main( ) {
```

```
    int x = 5;
```

```
    int y = 0;
```

```
    x = y;
```

```
    // هنا سيقوم بنقل الموجود فى y الى x
```

```
    printf("%d %d", x, y );
```

```
}
```



output:

0 0

[\(قم بتجربة الكود بنفسك واضغط هنا \)](#)

لكن إذا عكس المتغيرات كالتالي :

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

```
int main( ) {
```

```
    int x = 5;
```

```
    int y = 0;
```

```
    y = x;
```

```
    // هنا سيقوم بنقل الموجود في x إلى y
```

```
    printf("%d %d", x, y );
```

```
}
```

output:

5 5

[\(قم بتجربة الكود بنفسك واضغط هنا \)](#)

خطأ آخر :

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

```
int main( ) {
```

```
    int i, sum;
```

```
    for (i = 1; i <= 5; i++)
```

```
        sum += i;
```

```
    // نقوم بجمع الأرقام من واحد إلى خمسة
```

```
    printf("%d\n", sum);
```



}

output:

32782

(قم بتجربة الكود بنفسك واضغط هنا)

هنا كان يجب أن يقوم مجموع sum يساوي 15

لكن هنا لماذا يحدث ذلك؟

لأن متغير sum غير موجود به أى قيمة، عندما قمنا بتعريفه ولم نعطي له قيمة، أصبح يحتوي على أرقام من مخلفات البرامج السابقة يسمى هنا garbage لذلك لابد من تعريفه بصفر

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

```
int main( ) {
```

```
    int i, sum=0;
```

```
    for (i = 1; i <= 5; i++)
```

```
        sum += i;
```

```
    // نقوم بجمع الأرقام من واحد إلى خمسة
```

```
    printf("%d\n", sum);
```

}

output:

15

(قم بتجربة الكود بنفسك واضغط هنا)

خطأ آخر :

الأقواس { }

أمر **for** وأمر **if** يؤثر على سطر واحد فقط

لذلك إذا احتجنا إلى تنفيذ عدة أوامر فلا بد أن نضع أقواس { } إما إذا كان هناك أمر واحد



فلسنا بحاجة إلى كتابة الأقواس ، هناك بعض الناس عند كتابة الأقواس { } و هو بدون الحاجة إليها لأنه يؤثر على أمر واحد فقط ، يقوم بالنسيان ويضع أمر آخر معه هكذا :

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

```
int main( ) {
```

```
    int i, sum=0;
```

```
    for (i = 1; i <= 5; i++){
```

```
        sum += i;
```

```
        printf("%d\n", sum);
```

```
    }
```

```
}
```

output:

1

3

6

10

15

(قم بتجربة الكود بنفسك واضغط هنا)

لذلك إذا كان هناك أمر واحد فقط، تجنباً للخطأ لا نضع الأقواس، عندما نجد أمر الطباعة أو أي أمر تكرر أكثر من مرة ونحن نريده مرة واحدة فقط سنجد انه تم وضعه داخل أمر **for** لذلك ننظر إلى الأقواس

خطأ آخر شائع جداً :

إذا كان لدينا 5 أرقام ونريد طباعة أكبر رقم فيهم

في هذا السؤال سيحدث أخطاء كثيرة إذا كنت لا تدرب كثيراً
دعنا نرى أول خطأ :



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

```
int main( ) {  
    int i, max=0,x;  
    for (i = 1; i <= 5; i++) {  
        printf("Enter a number : ");  
        scanf("%d", &x);  
        if (x > max)  
            max = x;  
        // max إذا كان أول رقم أكبر من max إذا سجله داخل  
        // max إذا كان ثاني رقم أكبر من max إذا سجله داخل  
        // max إذا كان ثالث رقم أكبر من max إذا سجله داخل  
        // max وبالتالي سيكون هنا أكبر رقم مسجل في max  
    }  
    printf("%d", max);  
}
```

input:

5

3

9

7

200

output:

200

(قم بتجربة الكود بنفسك واضغط هنا)

لكن هذا البرنامج إذا قمت بحله بهذه الطريقة فى أى مقابلة عمل أو امتحان سيكون الناتج



خطأ، لأننا افترضنا أن الأرقام موجبة فقط، لكن ماذا إذا كان هناك أرقام سالبة فقط
مثال :

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

```
int main( ) {  
    int i, max=0,x;  
    for (i = 1; i <= 5; i++) {  
        printf("Enter a number : ");  
        scanf("%d", &x);  
        if (x > max)  
            max = x;  
        // max إذا كان أول رقم أكبر من max إذا سجله داخل  
        // max إذا كان ثاني رقم أكبر من max إذا سجله داخل  
        // max إذا كان ثالث رقم أكبر من max إذا سجله داخل  
        // max وبالتالي سيكون هنا أكبر رقم مسجل في max  
    }  
    printf("%d", max);  
}
```

input:

-1
-8
-6
-7
-2

output:

0



(قم بتجربة الكود بنفسك واضغط هنا)

لأن هنا قمنا بحجز max بصفر، ولذلك يجب فرض كل الاحتمالات عند عمل أى برنامج تجنباً للأخطاء، ومحاولة حل الكثير من التمارين والأسئلة، لأن البرمجة 10% تعليم و 90% ممارسة، ولذلك إذا كنت تقوم بقراءة الدرس ساعة فعليك ب 9 ساعات حل لتمرين وعمل برامج لم تدرسها سابقاً لتتفوق فى مجالك

(سنتعلم الطريقة الصحيحة لعمل هذا البرنامج في دروس متقدمة و لكن يمكنك التفكير حالياً في الحل الأمثل لهذا البرنامج و تختبر مهاراتك !)