

c programming

Written By The-oNe

Part 5

بسم الله الرحمن الرحيم

• الإتيافية:

لقد كتبت هذا الملف لغرض تثقيف الشباب العربي في مجال علوم الحاسب الآلي. وهذا الملف مجاني للجميع ولا أريد من ورائه إلا شيء بسيط جداً وهو دعوة صالحة في ظهر الغيب لي ولجميع أخواننا المسلمين في أنحاء العالم. كما أرجوا أن لا يتم التعديل في هذا الملف وإنسابه إلى غيري لأنني قد تعبت فيه كثيراً. كما أنني أوكد أنني أرحب وبكل سعة صدر بالنقد البناء الذي يستفيد منه الجميع. كما أنه إذا وجدت عزيزي القارئ أي أخطاء في هذا الملف يرجى أخباري بها وسيتم وضع أسمك في هذا الملف مع التعديل. كما أنني أرحب أن يشترك معي أي شخص لإضافة المزيد من الدروس لهذا الملف وسيتم وضع أسمه أيضاً وذلك حتى يكون هناك مرجع عربي للغة السي.

• جمل التكرار:

• جملة التكرار (for):

في بعض الأحيان نحتاج أن نقوم بعمل معين عدد معين من المرات دون تغيير وحيث أنه من الصعب تنفيذ هذا العمل إذا كان عدد مرات التكرار كبيرة جداً لذلك وجد البديل في لغة السي لهذا التكرار حيث عندما نستعين بهذه الجملة فإنها سوف تقوم بتكرار المهمة ذاتها عدد معين من المرات وهذا البديل هو استخدام الجملة (for). ويمكننا أن نقول أن هذه الجملة تقوم بتكرار مهمة معينة يحددها المبرمج أو المستخدم للبرنامج. والصورة العامة لهذه الجملة هو:

```
For (expression1 ; expression2 ; expression3)  
Statement;
```

حيث أن الجزء (expression1) يمثل نقطة البدء لعملية التكرار الذي تود القيام بها. أما (expression2) فهي تمثل نقطة التحقق أي هل سيتم إكمال هذا التكرار أما أن التكرار قد إنتهى. أما (expression3) فهي تمثل مقدار الزيادة التي سيتم فيها كل مرة الزيادة. ولاحظ أنه يتم الفصل بين هذه الأجزاء الثلاثة بفاصلة منقوطة (:). ونجد في النهاية ما يسمى ب (statement) وهي تعني الجملة التي ستنفذ في كل مرة تتم عملية التكرار والإفتراضي وجود جملة واحد فقط

ولكن يمكننا إضافة أكثر من جملة وذلك من خلال حصرها بين علامتين ({ }). ويمكننا معرفة ميكانيكية عمل الجملة (for) من خلال الخطوات التالية:

١. عند يأتي الدور على تنفيذ جملة (for) في البرنامج فإنه سيبدأ ب (expression1) حتى يرى القيمة الابتدائية لهذا التكرار ويسند هذه القيمة للمتغير المطلوب.

٢. بعد أن يحصل المفسر على نقطة البدء فإنه يتجه إلى (expression) حيث أنه هنا يقوم بالتحقق من قيمة هذا المتغير هل هي تحقق الشرط المطلوب أما لا أي أن حلقة التكرار لن يتم تكرارها إذا كان ناتج التحقق هو خطأ .

٣. بعد التحقق من الشرط يتجه البرنامج إلى تنفيذ ال (statement) ولاحظ أن (expression3) لا يتم النظر إليها حالياً .

٤. بعد تنفيذ ال (statement) يتجه البرنامج إلى (expression3) حيث يتم الزيادة أو النقص وذلك على حسب البرنامج.

٥. بعد تنفيذ (expression3) يتجه البرنامج إلى (expression2) وذلك حتى يتم التحقق هل القيمة الحالية لهذا المتغير تحقق الشرط فإذا كان صحيحاً نفذت ال (statement) وإذا كانت خطأ يخرج البرنامج من حلقة التكرار (for). لاحظ معي البرنامج التالي حتى يتم فهم هذه الحلقة التكرارية.

```
#include<stdio.h>
main () /* عشر (hello world) في هذا المثال سوف نقوم بطباعة الجملة */
/*مرات*/
{
int a;
for (a=1 ; a<=10 ;a++)
printf("hello world\n");
return 0;
}
```

في هذا المثال سوف يتم طباعة الجملة (hello world) عشر مرات. كما هو واضح في التعليق . ولاحظ أنه يمكنك أن تجعل البداية لحلقة التكرار والنهاية معتمدة على رغبة المستخدم لاحظ معي البرنامج الذي يقوم بعملية توليد أرقام متسلسلة .

```
#include<stdio.h>
main () /* هذا البرنامج يقوم بتوليد أعداد متسلسلة */
{
int num1,num2,count;
printf("enter start point:\n");
```

```
scanf("%d",&num1);
printf("enter the end point:\n");
scanf("%d",&num2);
for (count=num1 ;count<=num2 ;count++)
/* في السطر السابق استخدمنا الشرط (>=) وذلك للتحقق هل وصل البرنامج
لنقطة النهاية أما لا
printf("%d\n",count);
return 0;
}
```

لاحظ في البرنامج السابق أنه سوف يقوم بسرد جميع الأرقام المحصورة بين العددين المدخلين. وهنا من الجيد ذكر أن البرنامج السابق لا يعد برنامجاً كاملاً وذلك لأننا لو قمنا بإدخال العدد الأول أكبر من العدد الثاني فإن البرنامج سوف يتوقف عن العمل لذلك هنا يفيدنا ما تعلمنا سابقاً عن الجمل الشرطية لاحظ المثال التالي بعد أن قمنا بعملية إدخال الجملة الشرطية للبرنامج حتى نتحقق هل العدد الأول أقل من العدد الثاني أما لا.

```
#include<stdio.h>
main () /* هذا البرنامج يقوم بتوليد أعداد متسلسلة */
{
int num1,num2,count;
printf("enter start point:\n");
scanf("%d",&num1);
printf("enter the end point:\n");
scanf("%d",&num2);
if (num2> num1) /* هنا يتم التحقق هل هذا العدد الثاني أكبر من العدد
الأول فإذا
*/ (else) كان كذلك فإنه ينفذ بقية البرنامج وإلا فإنه سوف يتجه إلى الجملة
{
for (count=num1 ;count<=num2 ;count++)
/* في السطر السابق استخدمنا الشرط (>=) وذلك للتحقق هل وصل البرنامج
لنقطة النهاية أما لا
printf("%d\n",count);
}
```

```

else
printf("sorry i can`t do this loop");
return 0;
}

```

لاحظ في البرنامج السابق مدى الإستفادة العظيمة من الجمل الشرطية.

• جملة التكرار (while):

وهذه الجملة هي أحد الجملتين الثانية من جمل التكرار في لغة السي وتختلف هذه الجملة عن الجملة السابقة وهي (for) في أن جملة التكرار (for) لا نستطيع أن نعرف كما عدد المرات التي سيتم فيها التكرار ولكن في الجملة (while) فإن نعرف عدد المرات التي سوف نقوم بها بعملية التكرار . وفي الحقيقة هذه الجملة كثيراً ما تستعمل في جعل البرنامج لا منتهي إلا بتحقيق شرط معين كما سنرى في الأمثلة وطريقة عمل هذه الحلقة التكرارية أنها تقوم أولاً بفحص شرط التحقق فإذا تحقق الشرط فإنه تنفذ المطلوب من هذه الحلقة أما إذا كانت نتيجة التحقق غير صحيحة فإنها لا تنفذ الجمل التابعة لهذه الحلقة التكرارية وتعود إلى تكملة البرنامج من بعد الحلقة التكرارية. والصورة العامة لهذه الجملة هي :

```

While (expression)
Statement;

```

ولاحظ أنه بعد الجملة (while) تأتي جملة واحدة فقط (statement) ولكن يمكن أيضاً أن يكون أكثر من (statement) وذلك من خلال حصرها بين ({ }) . لاحظ البرنامج التالي وهو عبارة عن البرنامج الذي كتبناه من قبل وهو برنامج توليد الأعداد التسلسلية.

```

#include<stdio.h>
#include<stdlib.h> /* إستدعينا هذا الملف لأننا استخدمنا الدالة
(exit(0)) في

*/ البرنامج وذلك لعملية الخروج من البرنامج
main () /* هذا البرنامج يقوم بتوليد أعداد متسلسلة */
{
int num1,num2,count;
char ch;
printf("press any key to continue or press q to quit\n");
scanf("%c",&ch);
while (ch !='q') /* هنا سيتم الفحص هل الخيار الذي أدخلته هو الحرف

```

```

/* أما لا (q)
{
printf("enter start point:\n");
scanf("%d",&num1);
printf("enter the end point:\n");
scanf("%d",&num2);
if (num2> num1) /* هنا يتم التحقق هل هذا العدد الثاني أكبر من العدد
الأول فإذا

*/ (else) كان كذلك فإنه ينفذ بقية البرنامج وإلا فإنه سوف يتجه إلى الجملة
{
for (count=num1 ;count<=num2 ;count++)
في السطر السابق استخدمنا الشرط (>=) وذلك للتحقق هل وصل البرنامج
*/
لنقطة النهاية أما لا
printf("%d\n",count);
}
else
printf("sorry i can`t do this loop");
}
exit (0);/* تستخدم هذه الدالة للخروج من البرنامج
return 0;
}

```

لاحظ في البرنامج السابق لو أدخلت أي حرف من لوحة المفاتيح سوف يتم تنفيذ البرنامج ولكن إذا أدخلت الحرف (q) فإنه سوف يخرجك من البرنامج مباشرة. ولاحظ أن هذا البرنامج سوف يسألك مرة واحدة فقط ثم بعد ذلك بعد كل نهاية لعملية تكرار سوف يتم إعادة البرنامج نفسه وإن ضغطت على حرف (q) سوف يتم عمل تكرار لجمل البرنامج دون أي تدخل منك وبصورة سريعة وتستطيع إيقافها من خلال الضغط على (ctrl+c) نلاحظ هنا أننا واجهنا مشكلة بسيطة وهي أننا لا نستطيع عمل برنامج متكامل نستطيع استخدامه ومن ثم إغلاقه بصورة صحيحة. وسنتعلم باذن الله كيف نتغلب على هذه المشكلة.

• حلقة التكرار (while >----do):

وهذه الحلقة هي أحد الحلقات الموجودة في لغة السي وهي شبيهة إلى حد ما بحلقة التكرار (while) ولكن الفرق الوحيد بين هاتين الحلقتين في أن حلقة التكرار (

(while) من الممكن أن لا يتم تنفيذه أبداً وذلك عندما لا يتحقق الشرط بينما حلقة التكرار (do -----> while) يتم تنفيذهما مرة واحدة على الأقل وذلك لأن الشرط لا يتم تنفيذه إلا بعد عملية تنفيذ البرنامج. والصورة العامة لهذه الحلقة التكرارية هي :

```
Do  
Statement;  
While (expression);
```

نلاحظ أن الصورة السابقة هي الصورة العامة لهذه الحلقة التكرارية وهي تستطيع تنفيذ جملة واحدة فقط ولكنها أيضاً يمكنها القيام بتنفيذ أكثر من جملة وذلك على الصورة التالية :

```
Do  
{  
statement1;  
statement2;  
}  
while (expression);
```

وهنا يجب أن نتنبه أنه يوجد فاصلة منقوطة بعد (while(expression)) وتعمل هذه الحلقة التكرارية بأنها تقوم أولاً بتنفيذ الجمل التابعة للحلقة التكرارية وهي هنا (statement) ثم بعد الإنتهاء من هذه الجمل يتم التحقق من الشرط فإذا كان ناتج التحقق صحيح فإنه يعود وينفذ جمل احلقة التكرارية وإن كان خاطئاً فإنه يكمل عمل البرنامج. وفي الحقيقة هذه الدالة مفيدة جداً خاصة في برنامجنا الذي كتبناه سابقاً حيث أنه بإستخدام هذه الحلقة التكرارية نستطيع أن نتغلب على التكرار الغير منتهي كما حصل معنا عندما إستخدمنا الحلقة (while). لاحظ المثال التالي بعد إستخدام هذه الحلقة التكرارية.

```
#include<stdio.h>  
#include<stdlib.h> /* إستدعينا هذا الملف لأننا استخدمنا الدالة  
في البرنامج وذلك لعملية الخروج من (exit(0))  
  
*/  
البرنامج  
main () /* هذا البرنامج يقوم بتوليد أعداد متسلسلة */  
{  
int num1,num2,count;  
char ch;  
do  
{
```

```

printf("enter start point:\n");
scanf("%d",&num1);
printf("enter the end point:\n");
scanf("%d",&num2);
if (num2> num1) /* هنا يتم التحقق هل هذا العدد الثاني أكبر من العدد
الأول فإذا كان كذلك فإنه ينفذ بقية البرنامج

*/
else) /*
{
for (count=num1 ;count<=num2 ;count++)/* في السطر
السابق استخدمنا الشرط (>=) وذلك للتحقق هل

*/
printf("%d\n",count);
printf("press any key to continue or press q to quit\n");
scanf("%c",&ch);
}
else
printf("sorry i can`t do this loop");
printf("press any key to continue or press q to quit\n");
scanf("%c",&ch);
}
while (ch !='q'); /* هنا سيتم الفحص هل الخيار الذي أدخلته هو الحرف
(q) أما لا */
exit (0); /* تستخدم هذه الدالة للخروج من البرنامج */
return 0;
}

```

في البرنامج السابق استطعنا على التغلب من عملية التكرار اللا منتهي. وبهذا نكون قد إنتهينا من الحلقة التكرارية (do -----> while) .

• الجملة (break):

وتستخدم هذه الجملة عادة لكي نقوم بإيقاف عملية ما مثلاً لو أردت إيقاف عملية تكرار دون الوصول لنقطة النهاية فهذه الكلمة تقوم بطلبك بالتأكيد. لاحظ البرنامج التالي:

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

```

main () /* إستخدام الجملة (break) */
{
int num1;
for (num1=0 ; num1<=20 ; num1++)
{
printf("%d\n",num1);
لا حظ الشرط هنا إذا تحقق فإنه سوف يتم إيقاف عمل
/* البرنامج
break;
}
return 0;
}

```

لاحظ في البرنامج السابق أنه من المفترض أن يقوم البرنامج بسرد الأعداد من العدد (0) إلى العدد (20) ولكنه في الحقيقة لن يتم ذلك هنا وذلك لأننا إستخدمنا الجملة (break) حيث أنه سيتم طباعة الأعداد (1-10) وذلك لأنه شرط توقفنا أنه يساوي العدد (10).

ويوجد هناك أيضاً جملتين وهما (continue) والجملة (goto) ونظراً لقلة إستخدام الجملة (continue) فإني لم أجد أي مثال في أي كتاب عندي أما الجملة (goto) فغالباً مبرمجي لا ينصحون بإستخدام هذه الجملة في البرنامج لأن برامج عبارة عن متسلسلة بنوية أي كل جزء في البرنامج يكمل الجزء الذي يليه لذلك يفضل أن تكون برامج خالية من هذه الجملة كما أن هذه الجملة من الممكن أن تتسبب بأخطاء في برنامجك.

وتقبلوا خالص تحيات أخوكم المحب (The-oNe)
الرجاء إرسال مقترحاتكم وآرائكم على العناوين التالية:

The-one@pharaonics.net

OR

The_o0ne@hotmail.com

OR

The_o0ne@yahoo.com