c programming
Written By The-oNe
Part 4

بسم الله الرحمن الرحيم

- الإتفاقية:

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 لأني قد تعبت فيه كثيراً. كما أنني أؤكد 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 أني أرحب أن يشترك معي أي شخص لإضافة المزيد من الدروس لهذا الملف وسيتم وضع أسمه أيضاً وذلك حتى يكون هناك مرجع عربي للغة السي.

نظراً لأن أكثر البرنامج مبنية على التفاعل مع المستخدم فأحببت قبل البدء في الجمل الشرطية وجمل التكرار مناقشة دالتين للدخل والخرج وذلك حتى يتسنى لجميع فهم البرامج المكتوبة جيداً. وسيتم مناقشة هذه الدوال لاحقاً في أقسام أخرى من هذا السلسلة.

- إستخدام (printf):

تستخدم هذه الدالة وهي في الحقيقة ليست بدالة ولكنها أحد الكلمات المحجوزة للغة السي حيث أن لكل لغة برمجة كلماتها المحجوزة الخاصة بها. وتستخدم هذه الكلمة لطباعة شيء ما على شاشة الكمبيوتر.وهي تدعم موصفات الأعداد التي تمت مناقشتها سابقاً. وقد تعاملنا معها من قبل لذلك لا حاجة لذكر مثال عليها .

- إستخدام (scanf):

وهذه الكلمة أيضاً من الكلمات المحجوزة في لغة السي وهي تستخدم لأخذ المعلومات من المستخدم حتى يتم تنفيذ العلمية المطلوبة . والمثال التالي يبين كيفية التعامل مع هذا الدالة.

```
#include <stdio.h>
main ()
{
int a,b,c;
```

```
printf("enter tow number :=> \n");
scanf("%d \n",&a);
scanf("%d \n",&b);
c=a+b;
printf("%d",c);
return 0;
}
```

المثال السابق يقوم بأخذ عددين صحيحين من المستخدم ثم يقوم بجمع هذين العددين وطبع الناتج على الشاشة.والجديد فقط في هذا الكود هي السطر السادس والسابع والثامن لذلك هذا فقط الذي ساقوم بشرحه أما الباقي فقد تم شرحه فإن لم تفهم بقية الكود فإنه ينبغي عليك العودة لقراءة الملف من البداية. في السطر السادس استخدمنا الكلمة (scanf("%d",&a);) أولاً استخدمنا الكلمة (scanf) ثم بعد ذلك العلامة (%d) وذلك لأن المتغير الذي نتعامل معه هو من النوع العددي الصحيح (integer) أما العلامة (&a) فهي أنه عند إستخدام الدالة (scanf) لابد من إستخدام هذه العلامة ومعناها أن المفسر عندما يجد العلامة (&) متبوعة بمتغير ما فإن هذه العلامة ترمز إلى عنوان الحيز من الذاكرة أي أنه إذا لم يتم ذكر (&) عند إستخدام هذه الدالة فإن البرنامج لابد أن يعطي رسالة خطأ وذلك لأنه لا يعرف عنوان المكان الذي سيضع فيه القيمة المدخلة. أما السطر السابع فهو شبيه بالسطر السادس وهنا من الجيد ذكر أنه يمكن أخذ أكثر من مدخل من المستخدم في وقت واحد أي اننا في مثالنا هذا قد أخذنا عددين من المستخدم لذلك نستطيع أن نستخدم (scanf) بالصورة التالية لكي يتم أخذ قيم المتغيرين وإسنادها إليها .والصورة هي:

```
Scanf("%d %d ", &a,&b);
```

ويمكن لهذه الدالة أن تأخذ جميع أنواع المتغيرات ولكن ينبغي الإنتباه إلى نوع المتغير حتى تأخذ الهيئة التي تناسبه وأقصد بالهيئة أي إذا كان متغير عددي حقيقي فإننا نستخدم العلامة (%f) وإن كان المتغير حرفي فإننا نستخدم العلامة (%c) كما ذكر سابقاً.

- الجمل الشرطية:
- إستخدام (if).

وتستخدم هذه الجملة عندما يريد المبرمج أن يتأكد من شيء معين في برنامجه مثلاً هل قيمة متغير ما هي القيمة التي يريدها المبرمج أو المستخدم أم لا والصورة العامة لهذه الجملة هي.

```
If (expression)
Statement;
```

حيث أن (expression) هو الشرط الذي يراد التحقق منه أما (statement) فهي الجملة الذي سيتم تنفيذها في حال تحقق الشرط. ولاحظ أن علامة الفاصلة المنقوطة توضع فقط بعد (statement) ولايوضع بعد (if) أي فاصلة منقوطة. لاحظ المثال التالي.

```
#include<stdio.h>
main ()
{
int a;
printf("enter a number greater than 10:\n");
scanf("%d",&a); /* لابد من كتابة العلامة(&) حتى يتعرف البرنامج على عنوان التخزين*/
if (a <10) /* هنا يتحقق البرنامج هل هذا العدد أقل من عشرة أم لا فإذا كان أقل من ١٠ ستنفذ الجملة التالية وهي

عملية طباعة الجملة التالية*/
printf("the number you entered is not greater than 10\n");
return 0;
}
```

لاحظ في المثال السابق إستخدمنا الكلمة المحجوزة (scanf) لكي نقوم بأخذ قيمة ما من المستخدم ووضعها في المتغير (a).في هذا البرنامج كنا نريد من المستخدم أن يدخل لنا عدد أكبر من (10) لكن ماذا لو أدخل المستخدم عدداً أقل من (10) بالتأكيد سوف يؤثر هذا الإدخال في سير عمل البرنامج إذا كان البرنامج يعتمد على هذه القيمة. ولكن في برنامجنا هذا لا يهم العدد المدخل وذلك لعدم إتصال البرنامج بعمليات أخرى مبنية على هذا المدخل وإنما كان الهدف هو التحقق فقط هل هذا البرنامج أكبر من (10) أم لا. وكما قلنا سابقاً أن الجملة التي تلي الشرط تنفذ فقط في حال تحقق الشرط. لو أدخلنا للمثال السابق العدد (5) ولاحظ أنه أقل من العدد (10) أي أنه عند الجملة (if) سوف يكون شكل الشرط بعد التعويض عن القيمة (5<10) هل صحيح أن العدد (5) أقل من العدد (10) وبالطبع الناتج صح لذلك سوف يتحقق الشرط وسيتم تنفيذ الجملة التي تلي الجملة الشرطية وهي طباعة الجملة (the number you entered is not greater than 10) أي العدد المدخل ليس أقل من (10) بينما إذا أدخلنا عدداً أكبر من (10) لن يظهر شيء وذلك لأننا لم نحدد للبرنامج ماذا يفعل إذا كان العدد المدخل أكبر من (10). ولاحظ هنا أنه لو قمنا بإضافة جملة طباعة أخرى بعد جملة الطباعة السابقة سوف يتم تنفيذها سواءاً تحقق الشرط أم لا.

جرب البرنامج السابق وأدخل في المرة الأولى عدداً أقل من (10) وفي الأخرى أكبر من (10) ولاحظ الفرق بين الناتجين. ثم أضف الجملة (printf("welcome in my pro");) بعد جملة الطباعة السابقة ثم أعد تجربة البرنامج مرة أخرى. ولاحظ الفرق.

في الفقرة السابقة تعرفنا أن لغة السي تعترف فقط بجملة واحدة فقط بعد عملية الشرط. ولكنها أيضاً تعطي للمبرمج أن يستخدم أكثر من جملة بعد عملية الشرط وذلك على الصورة التالية:

```
If (expression)
{
statement1;
statement2;
.....
.....
statement n;
}
```

لاحظ أنه تستطيع أن تستخدم أكثر من جملة واحدة بعد العبارة الشرطية ولكن ينبغي أن تكون محصورة بين العلامتين ({ }). حاول تطبيق هذه العملية على البرنامج السابق ولاحظ الفرق.

- إستخدام الجملة (if ---- else):

وتقوم هذه العملية على أساس إذا كان الشرط صحيحاً فإفعل كذا وإن لم يكن صحيحاً فإعمل كذا والصورة العامة لها هي :

```
If (expression)
Statement1;
Else
Statement2;
```

ولاحظ أيضاً أنه لابد من جملة واحدة فقط بعد العملية الشرطية ويمكن حصر عدد من الجمل بين العلامتين ({ }) كما ذكرنا سابقاً . ولكي يتم فهم هذه العبارة جيداً نأخذ المثال السابق الذي قمنا به ونعدل فيه بحيث يكون إذا كان العدد المدخل أقل من العدد (10) يقوم بطباعة أن العدد أقل من (10) وإلا يتم طباعة أن العدد المدخل أكبر من (10).لاحظ المثال بعد التعديل:

```
#include<stdio.h>
main ()
{
int a;
```

```
printf("enter a number greater than 10:\n");
scanf("%d",&a); /* لابد من كتابة العلامة(&) حتى يتعرف البرنامج على عنوان التخزين*/
if (a <10) /* هنا يتحقق البرنامج هل هذا العدد أقل من عشرة أم لا فإذا كان أقل من ١٠ ستنفذ الجملة التالية وهي

عملية طباعة الجملة التالية*/
printf("the number you entered is not greater than 10\n");
else
printf("the number you entered is greater than 10\n");
return 0;
}
```

طبق البرنامج السابق وأنظر للنتائج بنفسك.

- الجمل الشرطية المتعددة (if elseif elseif else).

وتستخدم هذه الجملة عادة للتحقق من أكثر من شرط وذلك على الطريقة التالية:

```
If (expression)
Statement1;
Else
if (expression)
Statement2;
Else
Statement3;
```

وتستطيع أن تدخل در ما تشاء من الجمل الشرطية لاحظ المثال التالي الذي يقوم بفحص العدد المدخل هل هذا العدد ينتمي للأعداد في مدار معين أما لا.

```
#include<stdio.h>
main ()
{
int a;
printf("enter a number greater than 10:\n");
scanf("%d",&a);
if ( a>=0 && a<=10) /* إذا كان الشرطين صحيحين سوف تنفذ الجملة التي تليها راجع درس التعليمات لمعلومات

أكثر عن التعليمة (&&)*/
```

```
printf("the number you entered is between 0 and 10:\n");
else
if (a>10 && a<=20) /* إذا كانت الجملتين صحيحتين تنفذ الجملة التي تليها وأيضاً راجع درس التعليمات لملعومات

أكثر عن التعليمة (&&)*/
printf("the number you entered is between 11 and 20:\n");
else printf("the number you entered is greater than 21.\n");
return 0;
}
```

- إستخدام الجملة (switch):

وتستخدم هذه الجملة عادةً عندما يكون برنامجك يقوم بأكثر من مهمة مثلاً برنامج للآلة الحاسبة يعمل جيداً في حين إستخدمنا هذه الطريقة معه. والصورة العامة لهذه الجملة هو :

```
Switch (expression)
{
case constant 1:
statements
break;
case constant2:
statements
break;
case constant n:
statements
break;
default:
statement;
}
```

من النظر إليها وللوهلة الأولى يعتقد أنها معقدة ولكن في حقيقة الأمر تعتبر هذه الجملة من أسهل الجمل في لغة السي. ونعني ب (case constant) أي رقم الحالة أو الخيار الذي إذا ما أدخل سوف يتم تنفيذ الجمل التي تليه (statement)

ثم بعد ذلك نجد الجملة (break) وهي تستخدم للخروج من هذه الجملة. لأن الغرض المطلوب قد تم تنفيذه لذلك لا يوجد حاجة لتفحص الحالات التالية. ثم نجد في النهاية الكلمة (default:) وهي تعني أنه في حال تفحص جميع الحالات المدرجة ولم يجد مطابقة فإنه يقوم بتنفيذ العملية التي تليه. ولكي تتضح الفكرة أكثر سنصنع برنامج آلة حاسبة تقوم فقط بجمع أو طرح عددين فقط ويمكنك أن تجعل البرنامج يقوم بجميع العمليات .لاحظ المثال التالي:

```
#include<stdio.h>
main ()
{
int num1,num2,result;/* حجزنا ثلاثة أعداد لكي نقوم بالعلمية الحسابية على عددين ونضع الناتج في الثالث*/
char ch; /* حجزنا هذا النوع من المتغيرات وذلك حتى يتم إختيار العملية الحسابية مثلاً علامة الجمع (+) تعتبر متغير حرفي*/
printf("enter + to add numbers:\n");/* اختر (+) لعملية الجمع */
printf("enter - to subtract numbers:\n");/* اختر (-) لعملية الطرح*/
scanf("%c",&ch); /*لنأخذ العملية التي يريدها المستخدم*/
switch (ch)
{
case '+':
printf("enter the first number:\n");
scanf("%d",&num1);/*يأخذ قيمة المتغير الأول*/
printf("enter the second number:\n");
scanf("%d",&num2);/*يأخذ قيمة المتغير الثاني*/
result=num1+num2;
printf("result=%d +%d = %d",num1,num2,result);
break;
case '-':
printf("enter the first number:\n");
scanf("%d",&num1);/*يأخذ قيمة المتغير الأول*/
printf("enter the second number:\n");
scanf("%d",&num2);/*يأخذ قيمة المتغير الثاني*/
result=num1-num2;
printf("result=%d -%d = %d",num1,num2,result);
```

```
break;
default:
printf("unknown option:\n");
break;
}
return 0;
}
```

لو حاولت هنا إدخال أي خيار آخر غير (+) و (-) فسوف تظهر لك الجملة (unknown option) أي أن الإحتمال الذي أدخلته غير معرف لدى البرنامج.
وبهذا نكون قد إنتهينا من جزئية الجمل الشرطية وأتمنى أن تكون مفهومة لدى الجميع.

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_o0one@yahoo.com