جـــامعة القدس المفتوحة

كلية التكنولوجيا والعلوم التطبيقية

الدلـــيل العملي
لمقرر معالجة البيانات 1187

إعـــــداد د. ماجـــــــد حمايل

2014/2013

نسخة رقم (1)

فهرس محتويـــــات الدليل

مقدمة الدليل

نقدم لك عزيزي الطالب الدليل العملي لمقرر معالجة البيانات رقم 1187، في محاولة لكلية التكنولوجيا والعلوم التطبيقية لإثراء مادة المقرر من خلال عرضها بأسلوب التعلم الذاتي، والانتقال بك من السهل إلى الصعب بأسلوب سلس آملين أن يساعدك هذا الدليل في فهم أساسيات لغة البرمجة C.

قسم هذا الدليل إلى ثمانية أقسام توضح في مجملها المعارف والمهارات البرمجية الضرورية لفهم واستيعاب وتطبيق أساسيات هذه اللغة، وللتسهيل عليك فقد تم تقسيم الدليل إلى ثمانية أقسام كل قسم بحاجة إلى لقاء صفي بالإضافة إلى 2-3 ساعات بالمتوسط من الدراسة والتطبيق من قبل الطالب. أذكرك عزيزي الطالب بأن الممارسة لعملية وتنفيذ البرامج(الأمثلة) الواردة فيه والأسئلة التي تليها سيساعدك في النجاح والتفوق.

أهداف الدليل

يهدف هذا الدليل إلى تحقيق الأهداف الآتية:

1. توضيح استخدام جملة أو دالة الإخراج ()printf.
2. استخدام دوال الإدخال والعمليات الحسابية.
3. استخدام جملة الدوران ()for.
4. استكشاف جمل الدوران while و Do while.
5. تطبيق جمل الدوران من خلال أمثلة عملية.
6. تطبيق استخدامات الجمل الشرطية.
7. توضيح استخدامات المصفوفات من خلال أمثلة عملية.
8. توضيح أساسيات الدوال.

متطلبات الدليل

1. توفر Compiler لغة C/C++ مثل Borland C++ Ver 5.02.
2. إذا توفر لديك خدمة الإنترنت بإمكانك استخدام الموقع الإلكتروني التالي لتنفيذ البرامج الواردة فيه http://www.compileonline.com/

القسم الأول: دالة الإخراج

الأهداف: يهدف هذا القسم إلى تعريفك بالأشكال المختلفة لاستخدامات الدالة ()printf.

مثال رقم (1):

نفذ البرنامج التالي ولاحظ النتيجة:

```
// This is My first C program
# include <stdio.h>
# include <conio.h>
main()
    {
     printf(" AL-QUDS IS THE CAPITAL OF PALESTINE");
     getche();
    }
```

مثال رقم (2):

نفذ البرنامج التالي ولاحظ النتيجة:

```
/*This is My second C program */

# include <stdio.h>
# include <conio.h>
main()
    {
     printf(" Al_Quds is The Capital Of Palestine \n");
     getch();
    }
```

ربما تبدو لك النتيجة واحدة في كلا البرنامجين ولكن في الحقيقة أن هناك ثلاثة فروق بينهما، أذكرها؟

مثال رقم (3):

نفذ البرنامج التالي ولاحظ النتيجة:

```
/*Using Prinft and new line  */
# include <stdio.h>
# include <conio.h>
main()
    {
     printf(" AL-QUDS\n IS THE CAPITAL OF\n PALESTINE\n");
     getch();
     }
```

ربما لاحظت من نتيجة البرنامج السابق استخدام \n وتعني سطراً جديد New line.

مثال رقم (4):

لطباعة الثوابت باستخدام ()printf، نفذ البرنامج التالي ولاحظ النتيجة:

```
/*Using Prinft for printing constants  */
# include <stdio.h>
# include <conio.h>
main()
    {
     printf("%s Open University was established in %d ", "AL-QUDS", 1991);
     getch();
     }
```

من خلال استيعابك للبرنامج السابق كيف يمكن أن نطبع حرف فقط، وما هي وظيفة %d.

تعلـــــــم

Escape sequences in C are (not limited to): \n (newline) , \t (tab) , \v (vertical tab) , \b (backspace) , \r (carriage return).

Format Specifiers in C are (not limited to):

%i or %d	Int
%c	Char
%f	float
%s	String

القسم الثاني: دوال الإدخال والعمليات الحسابية

يهدف هذا القسم إلى:

- **استخدام المتغيرات ودالة الإخراج ()printf والإدخال ()getche و ()scanf ، ()gets**
- **استخدام العمليات الحسابية.**

عزيزي الطالب: ربما لاحظت من خلال المثالات السابقة أن من استخدامات printf طباعة الثوابت العددية %d والسلاسل الرمزية String و حرف واحد باستخدام %c.

والمثال التالي يشبه المقال السابق من حيث النتيجة:

```
# include <stdio.h>
# include <conio.h>
main()
   {
    int year=1991;
     printf (" Alquds Open University was established in %d",year);
     getche();
   }
```

مثل هذه البرامج ليست ذات فائدة وهذا يقودنا إلى ما يعرف بالمتغيرات وطريقة تعريفها.

مثال رقم (1):

لنفرض أننا نرغب في كتابة برنامج بحيث يطلب من المستخدم إدخال اسم أي جامعة وسنة التأسيس وطباعة ما تم إدخاله. إليك الحل:

```
# include <conio.h>
 # include <stdio.h>
main()
  {
  char name [30 ];
  int year;
  printf("Enter The University Name: ");
  gets(name);
  printf( " Enter The Year ");
  scanf("%d",&year);
  printf("%s was established in %d", name,year);
  getche();
  }
```

لقد أدخلنا في هذا البرنامج بعض الدوال الجديدة ومنها استخدام جملة الإدخال gets للسلاسل الرمزية وأجرينا تعديلا على تعريف المتغير name. ماذا يعني الرقم [30]؟

مثال رقم (2): بناء على ما سبق أكتب برنامج يطلب من المستخدم إدخال العمر بالسنوات ويطبع عمره بالأيام.

```
# include <conio.h>
 # include <stdio.h>
main()
  {
  int years,days;
```

```
  printf("Enter age in Years: ");
  scanf("%d",&years);
  days=years*365;
  printf(" The age in days is %d ",days);
  getche();
   }
```

من خلال البرنامج السابق ربما أدركت ما يلي:

1. **أنه تم تعريف متغيرين من النوع الصحيح.**
2. **استخدام جملة الإدخال scanf والإشارة & (ampersand) والتي تعني عنوان موقع الذاكرة الذي يخزن فيه قيمة المتغير المدخلة.**
3. **العملية الحسابية (*) وتعني الضرب.**

على فرض أن المستخدم أدخل العمر كالتالي: 1.5 ما هي النتيجة التي حصلت عليها وهل هي صحيحة أم خاطئة؟ ولماذا؟ وما الحل؟

مثال رقم (3):
يهدف هذا المثال إلى توضيح مفهوم الزيادة والنقصان.
نفذ البرنامج التالي ولاحظ النتيجة:

```
# include <conio.h>
 # include <stdio.h>
main()
  {
  int a=10;

  printf("The value of a is equal to %d\n ", a);
  a=a+1;
    printf("The value of a after increment is equal to %d\n ", a);
  getche();
   }
```

من البرنامج ربما توصلت إلى أن الناتج هو :

```
The value of a is equal to 10
 The value of a after increment is equal to 11
```

وهذا يعني أن قيمة المتغير a الابتدائية هي 10 وتم زيادة 1 عليها وأصبحت 11.
قم باستبدال a=a+1; بالجملة a=a++; ومن ثم بـ a+=1; أو a++; ولاحظ النتائج.
قم باستبدال ما سبق بإشارة الطرح ماذا تلاحظ؟

تعلـــــــــم

Operators In C Programming
Arithmetic Operators
Increment and Decrement Operators
Assignment Operators
Relational Operators
Logical Operators
Conditional Operators
Bitwise Operators
Special Operators

Arithmetic Operators

Operator	Meaning Of Operator
+	addition or unary plus
-	subtraction or unary minus
*	multiplication
/	division
%	remainder after division(modulo division)

Assignment Operators

The most common assignment operator is =. This operator assigns the value in right side to the left side. For example:

```
var=5   //5 is assigned to var
a=c;    //value of c is assigned to a
5=c;    // Error! 5 is a constant.
```

Operator	Example	Same As
=	a=b	a=b
+=	a+=b	a=a+b
-=	a-=b	a=a-b
*=	a*=b	a=a*b
/=	a/=b	a=a/b
%=	a%=b	a=a%b

Relational Operator

Relational operators checks relationship between two operands. If the relation is true, it returns value 1 and if the relation is false, it returns value 0. For example:

```
a>b
```

Here, > is a relational operator. If a is greater than b, a>b returns 1 if not then, it returns 0.

Relational operators are used in decision making and loops in C programming.

Operator	Meaning Of Operator	Example
==	Equal to	5==3 returns false (0)
>	Greater than	5>3 returns true (1)
<	Less than	5<3 returns false (0)
!=	Not equal to	5!=3 returns true(1)
>=	Greater than or equal to	5>=3 returns true (1)
<=	Less than or equal to	5<=3 return false (0)

Logical Operators

Logical operators are used to combine expressions containing relation operators. In C, there are 3 logical operators:

Operator	Meaning Of Operator	Example
&&	Logial AND	If c=5 and d=2 then,((c==5) && (d>5)) returns false.
||	Logical OR	If c=5 and d=2 then, ((c==5) || (d>5)) returns true.
!	Logical NOT	If c=5 then, !(c==5) returns false.

Increment and decrement operators

In C, ++ and -- are called increment and decrement operators respectively. Both of these operators are unary operators, i.e, used on single operand. ++ adds 1 to operand and -- subtracts 1 to operand respectively. For example:

```
Let a=5 and b=10
a++;   //a becomes 6
a--;   //a becomes 5
++a;   //a becomes 6
--a;   //a becomes 5
```

Source: http://www.programiz.com/c-programming/c-operators

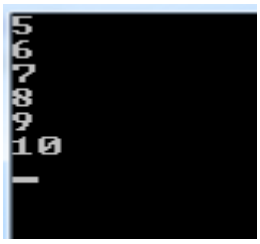
القسم الثالث: جملة الدوران for().

يهدف هذا اللقاء إلى : استخدام جملة الدوران for من خلال أمثلة عملية.

مثال (1) :عزيزي الطالب:إليك البرنامج التالي تم استخدام جملة الدوران for نفذه على حاسوبك وأجب عن الأسئلة الواردة أدناه:

```
# include <conio.h>
# include <stdio.h>
main()
{
 int N;
for (N=5;N<=10;N++)
printf("%d\n",N);
getche();
}
```

- **متى يتم تنفيذ الجملة التي تلي for؟**
- **استخدم الحرف N وهو حرف كبير فهل هذا ممكن في لغة c/c++؟ وضح.**
- **الناتج لهذا البرنامج هو:**



أي الأعداد من 5 إلى 10 بشكل عمودي، فهل لك أن تجري تعديلا بسيطا لطباعتها في سطر واحد؟ ثم استخدم \t بدل \n ماذا تلاحظ.

لو أدرنا طباعة الأرقام من 5 إلى 1000 فما التغيير على البرنامج الذي يحقق ذلك.

مثال (2) :اعتمادا على البرنامج السابق أكتب برنامج لطباعة كلمة Fabulous على الشاشة 50 مره.

ربما أدركت أن التغيير على البرنامج سيكون في بداية التكرار وشرطه وجملة الطباعة فقط.

مثال (3): يمكن استخدام جملة التكرار for لطباعة الأعداد الفردية والزوجية، والمثال التالي يوضح طباعة الأعداد الفردية من 1 إلى 12.

```
# include <conio.h>
```

```
# include <stdio.h>
main()
  {
   int odd;
    for( odd=1;odd<=12;odd++)
      {
    printf ("%d ",odd);
    odd=odd+1;
      }
     getche();
   }
```

مثال (4): بناء على ما سبق أعد كتابة البرنامج للأعداد الزوجية/ ثم أدرس البرنامج التالي وتعرف على الفرق بين البرنامج السابق واللاحق، أيهما أكثر فائدة.

```
#include<stdio.h>
#include <conio.h>
main()
{
   int number;
   int min,max;
    printf("Enter the minimum range: ");
   scanf("%d",&min);
   printf("Enter the maximum range: ");
   scanf("%d",&max);
   printf("Odd numbers in given range are: ");
   for(number = min;number <= max; number++)
       if(number % 2 !=0)
          printf("%d ",number);

   getche();
}
```

القسم الرابع: جمل الدوران while و do while

يهدف هذا اللقاء إلى :استخدام جملة الدوران while وجملة do while من خلال أمثلة عملية.

مثال (1) : لو رغبنا بإعادة كتابة البرنامج الوارد في المثال رقم (1)، – القسم الثالث– باستخدام while. إحدى الحلول تأخذ الشكل التالي.

```
#include<stdio.h>
#include <conio.h>
int main()
{
   int  N=5;
   while (N<= 10)
      {
         printf("%d\n",N);
          N=N+1;
      }
      getche();
}
```

- أعد كتابة البرنامج السابق لطباعة الأعداد الفردية من 19 – 7 .
- أكتب برنامج لطباعة الأعداد الزوجية بين 10 و 20؟

مثال (2): أعد كتابة البرنامج السابق باستخدام جملة الدوران do while؟

```
#include<stdio.h>
#include <conio.h>
int main()
{
   int  N=5;
   do
    {
     printf("%d\n",N);
     N=N+1;
     }
     while (N<=10);
      getche();
}
```

بناء على ما سبق:

- ما هو الفرق ما بين آلية عمل جملة while و do while.
- أعد كتابة البرنامج السابق لطباعة الأعداد الفردية التي تبدأ من 1 وتنهي ب 13.

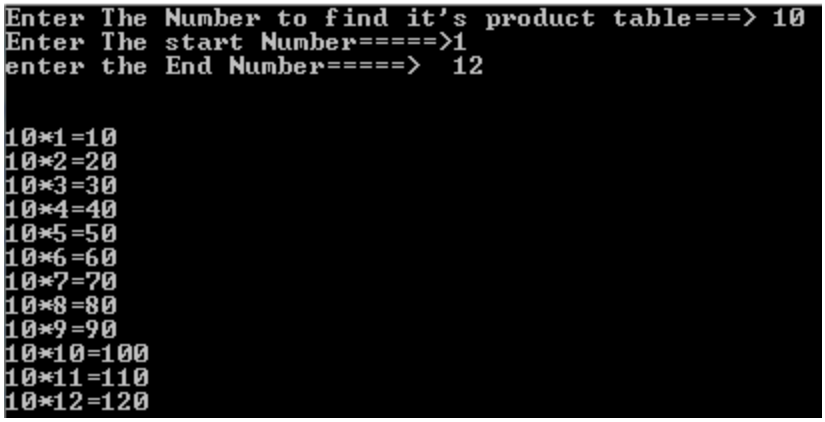
- **ما التغييرات التي تجريها على البرنامج السابق (طباعة الأعداد الفردية) ليطبع الأعداد الزوجية.**

القسم الخامس: تطبيقات متنوعة والتكرار المتداخل

مثال(1): أكتب برنامج لإيجاد جدول الضرب لأي عدد يدخله المستخدم كما يحدد المستخدم نقطة البداية والنهاية.

```
#include<stdio.h>
#include <conio.h>
int main()
{
   int N,F,L;
   int X,Y;
   printf("Enter The Number to find it's product table===> ");
   scanf("%d",&N);
   printf("Enter The start Number=====>");
   scanf("%d",&F);
   printf("enter the End Number=====>  ");
   scanf("%d",&L);
   printf("\n\n");
   for ( X=F; X<=L;X++)
    {
      Y=N*X;
      printf("%d" "*" "%d" "=" "%d\n",N,X,Y);
    }
      getche();
  }
```

لاحظ النتيجة :



بإمكانك عزيزي الطالب: إعادة تنفيذ البرنامج وإدخال أي عدد صحيح ونقطة البداية والنهاية للجدول.

- **أعد كتابة البرنامج السابق باستخدام while .**
- **أعد كتابة البرنامج السابق باستخدام do while.**

عزيزي الطالب: هناك مفهوم مهم وهو التكرار المتداخل nested Loop.

مثال (2): أكتب برنامج لطباعة جداول الضرب للأعداد من 1- 2 كل منها مضروبا من 1-10

```
#include<stdio.h>
#include <conio.h>
main()
{
   int N,F,R;
   printf("Welcome To The Power Of C/C++ Language\n");

      for (N=1;N<=2;N++)
       {
         for(F=1;F<=10;F++)
           {
            R=N*F;
            printf("%d" "*" "%d" "=" "%d\n" ,N,F,R);
            }
          printf("\n");
       }
      getche();
}
```

- اعتمادا على المثال السابق أعد كتابته بحيث يكون الناتج جداول الضرب للأعداد من 1-5 كل منها مضروبا من 1-7.

ملاحظة: لا يقتصر استخدام التكرار المتداخل على for ولكن يمكن استخدم while مع for وغيرها.

مثال(3): لتوضيح استخدام جمل الدوران إليك المثال التالي لايجاد مضروب أي عدد مدخل أكبر من الصفر. فمثلا مضروب 5! هو 5*4*3*2*1= 120

ويتوقف البرنامج عند إدخال الرقم صفر ويعطي مضروبه وهو 1.

```
#include<stdio.h>
#include <conio.h>
main()
{
   long N=1;
   long R;

   while (N !=0)
      {
       printf("Enter Number to find it's factorial=  ");
       scanf("%ld",&N);
       R=1;

        while( N>1)

            R=R*N--;
            printf("Factorial for given Number is= %ld\n",R);
       }
     getche();
}
```

القسم السادس: جمل الشرط

يهدف هذا اللقاء تعريفك باستخدام if و if else و switch.

مثال: أكتب برنامج للتعرف على الرقم المدخل هل هو أكبر من الصفر؟

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

```
   printf("Enter Any Number= ");
   scanf("%d",&N);
   if (N>0)
   printf("The Number you typed is greater Than zero");
   getche();
}
```

من المثال السابق يتضح أنه عند إدخال أي رقم أكبر من الصفر يطبع الرسالة أعلاه، ولكن ماذا يحدث عند إدخال رقم سالب أو صفر؟

مثال (2): نفذ البرنامج التالي الذي تم استخدام **if else** ولاحظ الفرق؟

```
#include<stdio.h>
#include <conio.h>
main()
{
   int N;
   printf("Enter Any Number= ");
   scanf("%d",&N);
   if (N>0)
   printf("The Number you typed is greater Than zero");
   else
   printf(" The Number You typed is equal or less than zero");
   getche();
}
```

ولو رغبنا في فحص الحالات الثلاث أكبر من صفر واقل من صفر، فإن البرنامج يصبح كالتالي:

```
#include<stdio.h>
#include <conio.h>
main()
{
   int N;
   printf("Enter Any Number= ");
   scanf("%d",&N);
   if (N>0)
   printf("The Number you typed is greater Than zero");
   else if (N<0)
   printf(" The Number You typed is  less than zero");
   else
   printf("The Number is equal to zero");
   getche();
}
```

مثال (3) : يوضح هذا المثال استخدام جملة if else أو أحيانا نسمية else if لإجراء العمليات الحسابية البسيطة +، *، / و – .

```
#include<stdio.h>
#include <conio.h>
main()
{
   float N1,N2;
   char op;

    printf(" Enter First Number operator and second Number ");
    scanf("%f %c  %f",&N1,&op,&N2);

       if (op== '+')
       printf(" The result is = %f", N1+N2);
       else
       if (op== '-')
       printf(" The result is = %f", N1-N2);
       else
       if (op== '*')
       printf(" The result is = %f", N1*N2);
       else
       if (op== '/')
       printf(" The result is = %f", N1/N2);
       getche();

      }
```

من المثال السابق تلاحظ صعوبة التركيب لمثل هذا البرنامج، خصوصا عندما تكون الخيارات كثيرة.

كما لاحظت أنه بعد انتهاء كل عملية علينا أن نعيد تنفيذ البرنامج، فالسؤال كيف لنا لمحافظة على استمرارية البرنامج ويكون المستخدم هو من يتحكم بإنهائه. إليك الحل.

مثال (4):

```
#include<stdio.h>
#include <conio.h>
main()
{
```

```
   float N1,N2;
   char op,ch;
    do
      {
       printf("\n Enter First Number operator and second Number ");
       scanf("%f %c  %f",&N1,&op,&N2);

       if (op== '+')
       printf(" The result is = %f", N1+N2);
       else
       if (op== '-')
       printf(" The result is = %f", N1-N2);
       else
       if (op== '*')
       printf(" The result is = %f", N1*N2);
       else
       if (op== '/')
       printf(" The result is = %f", N1/N2);

       printf ("\n\n Do You Want To Continue Using Calculator??? ");
        ch=getche();
        }
        while (ch== 'Y' || ch=='y');
        getche();
}
```

ملاحظة : يمكن إضافة الجمل الواردة في المثال السابق والتي تتيح للمستخدم إنهاء البرنامج، حاول تطبيق ذلك على الأمثلة الواردة في القسم الخامس.

مثال (5): يوضح هذا المثال استخدام جملة الانتقاء المتعدد switch، حاول فهم البرنامج جيدا، وأجب عن الأسئلة الواردة أدناه:

```
#include <stdio.h>
#include <conio.h>
main()
{
   int op;
   char ch;
   float no1,no2,r;
```

```
do
{
   printf("\n ********************  Simple Calculator********");
   printf("\n\n");
   printf("\n 1. Adding Numbers \n");
   printf("\n 2. Subtracting two Numbers \n");
   printf("\n 3.Multiplying two numbers \n\n");
   printf("\nEnter your option 1 or 2 or 3 \n");
   scanf("%d",&op);
   switch(op)
   {
   case 1:
         printf("Adding Numbers  \n");
         printf("-----------------\n");
         printf("Enter First Number=  ");
         scanf("%f",&no1);
         printf("Enter Second Number= ");
         scanf("%f",&no2);
         r=no1+no2;
         printf("The Result is=%f", r);
      break;
   case 2:
        printf("Subtracting two  Numbers\n");
        printf("----------------------\n ");
         printf("Enter First Number=  ");
         scanf("%f",&no1);
         printf("Enter Second Number=  ");
         scanf("%f",&no2);
         r=no1-no2;
          printf("The Result is=%f", r);

      break;
   case 3:
         printf("Multiplying  Numbers\n");
         printf("-------------------\n ");
         printf("Enter First Number ");
         scanf("%f",&no1);
         printf("Enter Second Number");
         scanf("%f",&no2);
```

```
            r=no1*no2;
             printf("The Result is=%f", r);

          break;

       }
       printf("\nDo you wish to continue[y/n]\n");
       ch=getche();
    }while(ch=='Y' || ch=='y');

    printf("\nPress any key to exit\n");
    getch();
}
```

بعد دراسة البرنامج السابق أجب عن الأسئلة التالية:

- ما هي الصيغة العامة لجملة switch.
- ما هي وظيفة getch().
- أضف خيارا رابعا لعملية القسمة.

القسم السابع: المصفوفات Arrays

يهدف هذا اللقاء إلى تعريفك بأهمية بطريقة بناء البرامج بلغة C/C++ المتعلقة بالمصفوفات ذات البعد الواحد وذات بعدين والعمليات الحسابية عليها.

مثال (1):

لو رغبنا بكتابة برنامج يطلب من المستخدم إدخال خمسة عناصر من الأعداد لصحيحة لمصفوفة أحادية (مكونة من 5 عناصر) وطباعة ما تم إدخاله على الشاشة؟

```
# include <stdio>
# include <conio>
main()

{
int mat[5];
int i;

printf("Enter 5 integer numbers== ");
```

```
for (i=0;i<5;i++)
{
  scanf("%d",&mat[i]);
}
 printf("The numbers you have entered are\n=== ");
 for (i=0;i<5;i++)
 {
  printf("%d\n",mat[i]);
}
 getche();
}
```

من خلال المثال السابق أجب عن الأسئلة التالية:

- **ما الشكل العام لتعريف المصفوفة الأحادية.**
- **ما الفرق بين استخدام جملة الإدخال لإدخال أنواع البيانات في المختلفة واستخدام طريقة المصفوفات من حيث تخزينها في الذاكرة.**

مثال (2): باستخدام المصفوفات أكتب برنامج لايجاد المجموع لخمس مقررات دراسية والمعدل العام لها؟

```
 # include <stdio>
 # include <conio>
 main()
 {
 float  mat[5];
 float sum=0;
 int i;
 float av;

 printf("Enter 5 integer numbers== \n");
 for (i=0;i<5;i++)
 {
   scanf("%f",&mat[i]);
   sum=sum+mat[i];
   av=sum/5;
 }
   printf(" The sum of the array elements = %f\n",sum);
   printf(" The average is = %f ",av);
   getche();
  }
```

من خلال المثال السابق، أجب عن الأسئلة التالية:

- **ما التعديل الذي تجرية على البرنامج السابق بحيث يجمع علامات 7 مقررات.**
- **ما التعديل الذي تجرية على البرنامج السابق بحيث يجمع علامات 12 مقررا.**
- **ما التعديل الذي تجريه على البرنامج أعلاه بحيث يطلب منك إدخال عدد المقررات ثم يجمع العلامات لها ويحسب المعدل ويطبعها على الشاشة.**

مثال (3):

نفذ البرنامج التالي موضحا مبدأ عمله.

```
#include <stdio.h>
#include <conio.h>
main()
{
 char name [20];
 float marks[5];
 int count;
 float ave;
 float sum=0;
 printf("enter the student name===== \n");
 gets(name);
 printf("enter marks for five subject=== \n");
 for (count=0;count<5;count++)
    {
    scanf("%f",&marks[count]);
    sum=sum+marks[count];
    }
     puts(name);
  for (count=0;count<5;count++)
      {
  printf("\n%f",marks[count]);
  }
  printf("\n%f",sum);
  ave=sum/5;
  printf("\n%.2f\n",ave);
  if (ave>=60)
  printf("Pass.....");
  else
  printf("fail.....");
```

```
  getche();
}
```

مثال (4): أكتب برنامج لجمع مصفوفتين من الرتبة الثانية؟

```
#include <stdio.h>
#include <conio.h>
void main()
{
   int m1[2][2],i,j,m2[2][2],add[2][2],r1=2,c1=2,r2=2,c2=2;

      printf("Enter rows and columns of First matrix \n");
      printf("Row wise please: \n");
      for(i=0;i<r1;i++)
      {
         for(j=0;j<c1;j++)
            scanf("%d",&m1[i][j]);
      }

      printf("Enter rows and columns of Second matrix \n");
      printf("Row wise\n");
      for(i=0;i<r2;i++)
      {
         for(j=0;j<c2;j++)
            scanf("%d",&m2[i][j]);
      }

         printf("Now we add both the above matrix \n");
         printf("The result of the addition is as follows;\n");
         for(i=0;i<r1;i++)
         {
            for(j=0;j<c1;j++)
            {
               add[i][j]=m1[i][j]+m2[i][j];
               printf("%d\t",add[i][j]);
            }
            printf("\n");
         }
  getche();
  }
```

اللقاء الثامن: الدوال Functions

يهدف هذا القسم إلى تعريفك بثلاثة أشكال لاستخدامات الدوال .

يستخدم مفهوم الدوال functions بطريقة كأنك تستخدم شخصا ما للقيام بعمل معين.

ويمثل المثال التالي أبسط أنواعه ويتكون من ثلاثة أقسام:

القسم الأول: قبل ()main وفيه نعرف الدالة باسمها منتهية بفاصله منقوطة.

القسم الثاني: function call وهو الذي يقوم باستدعاء الدالة الحقيقية وينفذها، لاحظ أنها تنتهي بفاصلة منقوطة.

القسم الثالث: وهي الدالة نفسها function it self مع ملاحظة أنها لا تنتهي بفاصلة منقوطة.

```
# include <stdio.h>
# include <conio.h>
numbers();    // function prototype
main()

{
  printf(" Welcome To The power of c\n");
 numbers();      //function call
printf("Thanks.....");
getche();
 }

 numbers()  // function defenition
    {
      int n1,n2,n3;
      printf("Enter First Number = ");
      scanf("%d",&n1);

     printf("Enter Second Number = ");
    scanf("%d",&n2);

     printf("Enter Third Number = ");
     scanf("%d",&n3);
     clrscr();
     gotoxy(10,10);
     printf( " The Numbers you entered are %d %d %d\n\n ",n1,n2,n3);
     getche();
     }
```

عزيزي الطالب: نفذ البرنامج السابق ووضح مبدأ عمله.

مثال (2): يوضح هذا المثال استخدام الدوال لإعادة قيمة باستخدام ()return.

```
# include <stdio.h>
# include <conio.h>
 int numbers();    // function prototype
main()

{
  int tot;
  printf(" Welcome To The power of c\n");

  tot= numbers();      //function call
  printf("The Total is = %d",tot);
  getche();

 }

 int numbers()  // function 25efinition
    {
      int n1,n2,n3;
      int sum;
      printf("Enter First Number = ");
      scanf("%d",&n1);

     printf("Enter Second Number = ");
    scanf("%d",&n2);

    printf("Enter Third Number = ");
     scanf("%d",&n3);
     clrscr();
     gotoxy(10,10);
     printf( " The Numbers you entered are %d %d %d\n ",n1,n2,n3);
     sum=n1+n2+n3;
     return (sum);
     getche();
          }
```

عزيزي الطالب:

- **نفذ البرنامج ووضح مبدأ عمله؟**
- **وضح ما هي أهمية استخدام هذا الشكل من البرمجة.**

القسم التاسع: أمثلة متنوعة

يهدف هذا القسم لتزويد الطالب بأمثلة إضافية وأسئلة لتعزيز فهم الجانب العملي من المقرر.

مثال(1): أكتب برنامج لقراءة اسمك وطباعته باستخدام gets() و puts().

```
# include <stdio.h>
# include <conio.h>
main()
{
 char name [30];
printf (" Enter your Name: ");
gets(name);

printf("Your Name is: ");
puts(name);
getche();

}
```

ما الفرق بين المثال السابق والبرنامج التالي:

```
# include <stdio.h>
# include <conio.h>
main()
{
 char name [30];
printf (" Enter your Name: ");
scanf("%s",name);

printf("Your Name is %s ", name);

getche();

}
```

ملاحظة: حاول إدخال اسمك الأول واسم العائلة في كلا البرنامجين ستلاحظ الفرق ومحدودية استخدام scanf في حالة استخدمها لطباعة متغيرات السلاسل الرمزية string.

مثال (2): البرنامج التالي اشتمل على دالتين جديدتين هما gotoxy() clrscr()

```
# include <stdio.h>
# include <conio.h>
main()

{
 int n1,n2,n3;

 printf("Enter First Number = ");
 scanf("%d",&n1);

 printf("Enter Second Number = ");
 scanf("%d",&n2);

 printf("Enter Third Number = ");
 scanf("%d",&n3);

 clrscr();
 gotoxy(10,10);
 printf( " The Numbers you entered are %d %d %d ",n1,n2,n3);
 getche();
  }
```

نفذ البرنامج ولاحظ النتيجة.

المراجع

http://www.programiz.com/c-programming.

http://aelinik.free.fr/c/

انتهى