



Milestone4U

فريق

الملخص الشامل

لمادة البرمجة I

إعداد الأنسة:

م. حنان حسين

إدارة الفريق:

م. حنان حسين



+963 988 211 754

منظور البرمجة :

هو منظور ثابت لا يتغير من لغة إلى لغة أخرى .

الخوارزمية :

هي عدد محدد من الخطوات المتتالية المنطقية التي تقود بالنتيجة إلى كهدف محدد أو حل مشكلة .

البرنامج الحاسوبي :

هو عبارة عن مجموعة التعليمات لمآلة صيغت بشكل خوارزمية .

صفات الخوارزمية :

- 1- المحدودية : أنها تحل عدد محدد من الخطوات .
- 2- الاستثنائية : كل خطوة من الخطوات تقدم نتائج مختلفة عن الخطوات الأخرى .
- 3- الدقة المنطقية والعلمية : يجب أن تكون الخوارزمية قابلة للتصديق بالمنظور العلمي .  
مثال : عند كتابة خوارزمية لإظهار عاملي عدد ما ( يجب أن تكون للأعداد الموجبة فقط ) .
- 4- تأخذ قيمة أو عدة قيم على أنها مدخلات (input) .
- 5- تأخذ قيمة أو عدة قيم على أنها مخرجات (output) .

شروط كتابة أي خوارزمية على الحاسب :

[1] أن تكون عامة : يجب كتابة الخوارزمية بشكل عام وبعدها يتم إضافة الحالات الخاصة التي تعالجها المآلة المدروسة .

[2] أن تكون مرنة : لا يجب كتابة الخوارزمية ككل متكامل ، وذلك لأن عملية البرمجة هي عملية تراكمية وقابلة للتطوير ، وذلك علينا كتابة الخوارزمية كوحدة مستقلة لأنه يسهل التعديل عليها دون إعادة بناء الخوارزمية من الصفر .

[3] يجب تعريف كافة أنواع المدخلات والمخرجات : وتكون المدخلات على عدة أنواع مثل (مخارج ، أرقام ، سلاسل ، ... )  
مثال : جذر عدد : المدخلات هي عدد صحيح أو حقيقي من المخرجات حقيقي دائماً .  
عملية القسمة : إذا كانت المدخلات صحيحة من المخرجات أعداد حقيقية قطعاً .

هيكلية البرنامج الكاسوي في #C :

```

using System;
namespace First_App;
{
class Program
{
static void Main(string[] args)
{
Console.WriteLine("Hello C#");
}
}
}
    
```

الدلالة على نظام الأوامر  
 مكتبة (فضاء أسماء) يقدم  
 مجموعة الصفوف المجهزة والمعرفة  
 التي يمكن استخدامها للتطوير مباشرة  
 الاشارة Main في الصف Program  
 يمكن اعتبارها نقطة انطلاق لتنفيذ البرنامج  
 Console.WriteLine هي الصنف في الصنف Console  
 تستخدم للتعامل مع  
 الواحدة النصية السوداء

- لغة #C كمنظمة التوجيه
- وبالتالي تعتمد على الصفوف class
- الصف: هو تجميع لمعطيات ووظائف (توابه وهرانيات).
- فضاء الأسماء namespace: هو تجميع لعدد من الصفوف

Hello C#

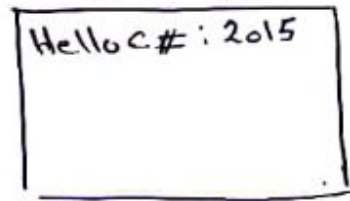
المتولات والأعاط الأساسية :

المتول: هو مكان في الذاكرة يُقدر بالبايت ويتعلمه بالنظر المستخدم.

الشرح	المساحة مقدره بالبايت	Data_Type
قيمة صحيحة بدون اشارة تراوح بين 0 و 255	1	byte
قيمة صحيحة ذات اشارة تراوح بين -128 و 127	1	sbyte
قيمة صحيحة ذات اشارة تراوح بين -2 <sup>15</sup> و 2 <sup>15</sup>	2	short
قيمة صحيحة بدون اشارة تراوح بين 0 و 2 <sup>16</sup>	2	ushort
قيمة صحيحة ذات اشارة تراوح بين -2 <sup>31</sup> و 2 <sup>31</sup>	4	int
قيمة صحيحة بدون اشارة تراوح بين 0 و 2 <sup>32</sup>	4	uint
قيمة صحيحة ذات اشارة تراوح بين -2 <sup>63</sup> و 2 <sup>63</sup>	8	long
قيمة صحيحة بدون اشارة تراوح بين 0 و 2 <sup>64</sup>	8	ulong
عدد عشري (فاصله عائمة) بدقة بسيطة 7 أرقام عشرية	4	float
عدد عشري، بدقة مضاعفة 15 رقم عشري	8	double
عدد عشري، بدقة مضاعفة 28 رقم عشري على الاقل	16	decimal
سلسلة محارف [ على مركبة ]	-	string
حرف	2	char
رأب في الذاكرة	-	bool

المخرج التنفيذي

```
int y=2015;  
string s="Hello c#:";  
console.WriteLine(s+y);
```



قواعد تسمية المتغيرات

- 1- تبدأ أسماء المتغيرات بحرف من الحروف اللاتينية (a-z) قليلاً اختيارياً سلسلة من الأرقام والأحرف.
- 2- ممنوع استخدام الرمز الخاصة أو الغريبة في أسماء المتغيرات ما عدا ( \_ ) الشوكة التحتية.
- 3- يجب ألا يكون + المتول مطابقاً لأي من الكلمات المحبزة للغة.
- 4- لغة C# تفرق بين الأضرب الكبيرة والصغيرة [ + المتول n هو كود المتول N ].

أمثلة عن تعريف المتغيرات

```
int MaxN=50;  
double Value2=2.3;  
char achar='k';  
bool Test=false;
```

```
int MaxN;  
double Value2;  
char achar;  
bool Test;  
Long N_F;
```

التوابيع

```
const int Num=0;  
X const int Num;
```

وتتبع عبارة اسناد التابيع Num (تغير قيمته بعد التعريف) يحدث خطأ أثناء  
عند عدم إمكانية تعديل محتواه.

• المترجم : يقوم بتحويل الكود المكتوب بلغة البرمجة إلى لغة الآلة (0, 1).

العمليات المنطقية:	العمليات المقارنة	العمليات الحسابية
الترتيب: 1 ! نفي	الترتيب: 5 > أكبر تماماً	الترتيب: 1 + إشارة موجبة
7 and &	5 >= أكبر أو يساوي	1 - إشارة سالبة
9 or	5 < أصغر تماماً	1 ++ زيادة بمقدار واحد
10 and &&	5 <= أصغر أو يساوي	1 -- نقصان بمقدار واحد
11 or	6 == يساوي	2 * عملية الضرب
	6 != لا يساوي	2 / عملية القسمة
		2 % عملية باقي القسمة
		3 + عملية الجمع
		3 - عملية الطرح

أفضليات العمليات:

- فك الأقواس إن وجدت.
- تنفيذ العمليات تبعاً للأفضليات.
- في حال تساوي عمليتين لهما نفس الأفضلية، فتكون الأسبقية للعمليات الموجودة على اليسار.

مثال العبارة  $(x+y > z) \&\& (z < 8)$

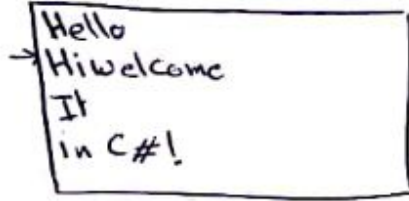
## تعليمة الكتابة / writeLine

تالعة للصف Console

```

console.WriteLine("Hello");
console.write("Hi");
console.WriteLine("welcome\n It\n in C#!");

```



## تعليمة القراءة / ReadLine

تالعة للصف Console ، ترفع قيمة فانزلة لة معرفة String

```

string s; int num;
console.WriteLine("enter the value :");
s = console.ReadLine();
num = Int32.Parse(s);

```

لكل نطاب بيط يوجد منه مقابل للاعرائية Parse

الصف	النط
Byte	byte
SByte	sbyte
Int16	short
Int16	ushort
Int32	int
Int32	uint
Int64	long
Int64	ulong
Single	float
Double	double
char	char

```

MessageBox.Show("welcom");

```

غير مسوع باستوار  
بوا حنة ال Console

1. string firstNumber, // first string entered by user
2. secondNumber;
3. int number1, number2, Sum;
4. Console.WriteLine("please enter the first integer:");
5. firstNumber = Console.ReadLine();
6. Console.WriteLine("\n please enter the second integer:");
7. secondNumber = Console.ReadLine();
8. number1 = Int32.Parse(firstNumber);
9. number2 = Int32.Parse(secondNumber);
10. Sum = number1 + number2;
11. Console.WriteLine("\n The sum " + Sum);

```

please enter the first integer : 5
please enter the second integer : 25
The sum 30
Press any key to continue
  
```

التمرير التنفيذي  
لقيم افتراضية

// تعليقات - سطرية .

... /\* تعليقات لأكثر من سطر .  
- - -  
- - -  
- - - \*

• التعليقات هي عبارات للإبارة المترجم ولا ينفذها . يكتبها المبرمجون لتوضيح تعليمات البرنامج ولتجعله أكثر قابلية للفهم والقراءة .

1) الإسناد البسيط:

الشكل العام للإسناد البسيط هو  $\boxed{\text{Variable\_Name} = \text{expression}}$  أو

- مثال:
- أسناد متعدد:  $x = y = c = d$ ;
  - تعبير:  $x = y + 2 * 10$ ;
  - متحول:  $x = y$ ;
  - قيمة:  $x = 5$ ;

2) الإسناد المزود بعملية:

الشكل العام  $\boxed{\text{Variable (operation)} = \text{expression}}$

مثال: نفرض  $y = 12$  و  $x = 1$  فإن نتائج العمليات الآتية لكل باء عدة هو:

- $x += y$  و  $y = 12$  ،  $x = 13$  //  $x = 1 + 12 = 13$   $\Rightarrow x = x + y$  و  $x = 1$
- $x * = y$  و  $y = 12$  ،  $x = 12$  //  $x = 1 * 12 = 12$   $\Rightarrow x = x * y$  و  $x = 1$

• عملية الجمع التقدمي ++x

تقوم بزيادة واحد  $x$  ، بحيث تزداد القيمة قبل استخراج قيمة التعبير.

• عملية الجمع التأخيري x++

تقوم بزيادة واحد  $x$  ، بحيث يتم استخراج قيمة التعبير أولاً وبعدها يتم زيادة  $x$ .

• عملية الطرح التقدمي x--

تقوم بطرح واحد من  $x$  ، بحيث يتناقص القيمة قبل استخراج قيمة التعبير.

• عملية الطرح التأخيري --x

تقوم بطرح واحد من  $x$  ، بحيث يتم استخراج قيمة التعبير أولاً وبعدها يتم تناقص  $x$ .

مثال: بفرض لدينا المتغيرات الصحيحة x, y, z والمطلوب إيجاد قيم المتغيرات بعد كل تعليمة.

int x, y, z;

x = y = z = 0; // x = 0    y = 0    z = 0

x = (++y) + (++z); // x = 2    y = 1    z = 1

x = (y++) + (z++); // x = 2    y = 2    z = 2

x = (++y) + (z++); // x = 5    y = 3    z = 3

x = (y--) + (--z); // x = 5    y = 2    z = 2

x = (--y) + (z--); // x = 3    y = 1    z = 1

تكملة:

السؤال الأول: اختر العبارة الصحيحة لكل مما يأتي:

1- التصريح عن متحول من نوع Integer وسادسية ابتدائية له نكتب

A - int x = "Hello";

B - int x = 0,5;

C - int x = 4;

2- نقطة بدء تنفيذ المشروع في # C هي الكلمة المحجوزة:

A - Begin

B - Main

C - begin

D - main

3- سيكون خرج السطر التالي و Console.WriteLine(123) هو:

A - 123 وسيغيرها كمنهج

B - 123 وسيغيرها كعدد

C - سيعطي رسالة خطأ

D - لن تقوم بطباعة الرسالة

4- سيكون فخرج السطر الآتي ز (123) Console.WriteLine هو:

- A - 123 وسيغيرها كنها
- B - 123 وسيغيرها كعدد
- C - رسالة خطأ
- D - لن تقوم بطباعة الرسالة

5- أي العبارات الآتية صحيحة برمجياً:

- A - String Console;
- B - float 14All;
- C - int my\_number;
- D - int Num = 4.5;

6- أي العبارات الآتية صحيحة برمجياً:

- A - String Val = 'hello';
- B - String Val = "hello";
- C - String Val = 25;
- D - كل ما ذكر صحيح

7- برمز العملية 'لا يساوي' في لغة #C بالرمز:

- A - <>
- B - ><
- C - !=
- D - =!

السؤال الثاني: ضع إشارة صح أو خطأ وسم الخطأ إن وجد.

- 1- عند استخدام العملية `Num = Console.ReadLine` فإن `Num` يكون من النوع `int`.
- 2- نستطيع التصريح عند ثابته كالتالي `Const int Num`.
- 3- `Parse` دالة أحد أشكال `Single.Parse`.
- 4- في لغة `C#` يمكن استخدام متغيرات غير مصرح عنها.
- 5- تستخدم عبارة `using` للتصريح عن المتولات.
- 6- نتجاهل التعليقات عند الترجمة في حال عدم وجود أخطاء في صياغتها.
- 7- إذا كان  $x = 3$  و  $y = 1$  والقيمة الناتجة  $z = 4$  فإننا نخطئ على رسالة خطأ.

السؤال الثالث:

أوجد ناتج تقييم التعبير البرمجي الآتي مستخدماً الأولويات لتنفيذ العمليات مع العلم أنه:

$$c = 7, b = 10, a = 2, x = 6$$

$$y = a * x * x + b * x + c$$

السؤال الرابع: اكتب برنامج بلغة `C#` يقوم بحساب مجموع خمسة أعداد صحيحة مدخلة من لوحة المفاتيح وطباعة الناتج.

السؤال الخامس، اكتب برنامج بلغة C يقوم بحساب مساحة ومحيط دائرة، يتم إدخال نصف قطرها  $r$  من لوحة المفاتيح وطباعة النتيجة.

```
static void Main(string[] args)
{
    const float p = 3.14f;
    float r;
    Console.WriteLine("Enter r = ");
    r = Single.Parse(Console.ReadLine());
    float a = p * r * r;
    Console.WriteLine("The area of circle is = " + a);
    float c = 2 * p * r;
    Console.WriteLine("The circumference of circle is = " + c);
}
```

الترجيع التنفيذي بفرض  $r=5$

```
Enter r = 5
The area of circle is = 78.5
The circumference of circle is = 31.4
```

السؤال السادس: اكتب برنامج بلغة C لإيجاد قيمة y حيث  
 $y = x^2 + 5x - 3$  كلما أن  $x = 10$  واطبع الناتج.

السؤال السابع: اكتب في المتولات بعد تنفيذ كل سطر من أسطر العمليات يدوياً دون كتابة برنامج لحاسبها.

	a	b	c	d	e
1- int a=0, b=0, c=8, d=0, e=0;	0	0	8	0	0
2- a=b=c;	8	8	8	0	0
3- b*=5;	8	40	8	0	0
4- a/=2;	4	40	8	0	0
5- e = ((a+b) * (d=6)) / 2;	4	40	8	6	132
6- d%=5;	4	40	8	1	132
7- a = b * c - d;	319	40	8	1	132
8- e = d = c = b = (a = (a % 31 - 9));	-2	-2	-2	-2	-2

مثال 2: أمين ستظهر الأخطاء الناجمة عن تعريف مدى المتولات في البرنامج التالي.

```
int a, b = 12;
{
    int x, y = 8;
    {
        int k = 0;
        x = a * y;
        k = k + 2; // Error
    }
}
{
    int z = 12;
    a = z + b;
    {
        int u = 0;
        x = a * y; // Error
        u++;
    }
}
{
    int u = 1;
    a = u - b;
    z = u - b; // Error
}
```

مثال 1

```
//Block0
int a, b = 12;

//Block1
{
    int x, y = 8;
}

//Block2
{
    int z = 12;
    a = b + z;
    x = z; // Error
    a = x + 1; // Error
}

//Block3
{
    int u = 1;
    a = b - u;
    y = u - b; // Error
}
```

• يظهر الخطأ نتيجة عدم وجود أي تعريف لكل من x, y في الكتل التي تظهر بها.  
• في حين لا توجد أي مشكلة في استخدام المتولات b, y في هذه الكتل.

• يمكن التصريح عن المتول بأي مكان بالبرنامج ليس بالضرورة ببداية البرنامج.

• في C لا يمكن استخدام نفس تعريف الاسم في الكتل المتداخلة إلا في حالة كتلة الصنف وكتلة التابع.

`if (condition) statement ;`

الاظمية الشرطية البسطة ، ان الكلام العام لا

واذا كانت الاظمية مركبة (أي مؤلفة من مجموعتين شرطيات) توضع بين قوسين كلمة

```

if (Condition)
{
    statement1;
    statement2;
    ---
    ---
}
    
```

كلمة Condition : الشرط وتنفذ معاملات مقارنته وتواجه منطقته ونتيجته True و False

• يتم اختبار الشرط فعندما يتحقق الشرط تنفذ الاظمية Statement وعند عدم تحقق الشرط تتجاهل الاظمية Statement وتنفذ الاظمية التي تأتي عبارة if .

مثال 1 : كتابة شفرة بلغة #C تقرأ علامة الطالب في أحد الامتحانات وتطبع كلمة Passed إذا كانت العلامة أكبر أو تساوي 60 فدرج بغيره الاذغال 80

```

1. float grad;
2. console.WriteLine("Enter the grad:");
3. grad = Int32.Parse(console.ReadLine());
4. if (grad >= 60)
5.     console.WriteLine("Passed");
    
```

```

Enter the grad:
80
Passed
    
```

```

Enter the grad:
40
    
```

مثال 2 : اكتب شفرة بلغة #C تقرأ عددين صحيحين (Num1, Num2) وترتبهما تصاعدياً بالتبديل بين قيمتهما (إذا كان الأول أكبر من الثاني يبدل بين قيمتهما)

```

1. int Num1, Num2, Num3;
2. console.WriteLine("Enter a number 1:");
3. Num1 = Int32.Parse(console.ReadLine());
4. console.WriteLine("Enter a number 2:");
5. Num2 = Int32.Parse(console.ReadLine());
6. if (Num1 > Num2)
7.     {
8.         Num3 = Num1;
9.         Num1 = Num2;
10.        Num2 = Num3;
11.    }
12. console.WriteLine("The new order is Num1=" + Num1 + " Num2=" + Num2);
    
```

```
if (condition) Statement1;  
else Statement2;
```

الشكل العام لها 1

- عندما يتحقق الشرط تُنفذ العبارة الأولى Statement1 وإذا لم يتحقق الشرط تُنفذ العبارة الثانية Statement2.
- وإذا كانت العملية مركبة قوضع بين { } أقواس كالتالي.

مثال 1: اكتب شيفرة تقرأ درجة طالب في أحد الامتحانات وتطبع كلمة Passed إذا كانت الدرجة أكبر أو تساوي 60 وكلمة Failed إذا كانت الدرجة أصغر تماماً من 60.

```
float grad; string sgrad;  
Console.WriteLine("Enter the grad:");  
sgrad = Console.ReadLine();  
grad = Single.Parse(sgrad);  
if (grad >= 60)  
    Console.WriteLine("Passed");  
else  
    Console.WriteLine("Failed");
```

```
Enter the grad:  
70  
Passed
```

```
Enter the grad:  
35  
Failed
```

مثال 2: اكتب شيفرة تقرأ عدداً حقيقياً (num) ثم تطبع كلمة Positive إذا كان العدد موجباً أو كلمة negative إذا كان العدد سلبياً.

مثال 3: اكتب شيفرة تقرأ عدداً صحيحاً (Value) وتطبع كلمة odd إذا كان العدد فردياً وكلمة even إذا كان العدد زوجياً.

1. int value;
2. Console.WriteLine("Enter integer number:");
3. Value = int.Parse(Console.ReadLine());
4. if (value % 2 == 0)
5.     Console.WriteLine("even");
6. else
7.     Console.WriteLine("odd");

Enter integer number: 25  
odd

Enter integer number: 16  
even

### [3] تعليمة if...else الموسعة (المتداخلة)

```

if ( شرط 1 )
    ; تعليمة 1
else if ( شرط 2 )
    ; تعليمة 2
...
else
    ; تعليمة 4
    
```

إذا تحقق الشرط 1 تُنفذ التعليمة 1 ويتجاهل باقي البنية  
وإذا لم يتحقق الشرط 1 يتجاهل التعليمة 1 وينتقل  
ليختبر الشرط 2 إذا كان محققاً تُنفذ تعليمة 2 ويتجاهل  
باقي البنية وهكذا...  
وعند وصوله لـ else وعدم تحققه أي شرط  
فإنه سينفذ تعليمة الـ else

مثال: اكتب شيفرة تقوم بحساب قيمة y كالتالي

$$y = \begin{cases} x+1 & \text{if } (x > 0) \\ x^2-3 & \text{if } (x = 0) \\ 2x-4 & \text{if } (x < 0) \end{cases}$$

```

int x, y;
console.WriteLine("Enter x = ");
x = Int32.Parse(console.ReadLine());
if (x > 0)
{
    y = x + 1;
    System.Console.WriteLine("y = " + y);
}
else
{
    if (x == 0)
    {
        y = x * x - 3;
        System.Console.WriteLine("y = " + y);
    }
    else
    {
        y = 2 * x + 4;
        console.WriteLine("y = " + y);
    }
}

```

التمرين التنفيذي بإدخال القيمة 4 هو:

Enter x = 4  
y = 5

تمرين: أوجد النتيجة

```

int a = 100, b = 0, c = 8, e = 100;
if (b == 0)
    b += 3;
if (a <= 0)
    a = a + b;
else
    a = a * b;
console.WriteLine("a = " + a);

```

a = 300

الشكل العام :

Condition ? Value<sub>1</sub> : Value<sub>2</sub> ;

• في حال تحقق الشرط سيتم ارجاع قيمة Value<sub>1</sub> وإلا سيتم ارجاع Value<sub>2</sub>

مثال 1 :

<p>مثال 3 :</p> <pre> c = ((a+b) &gt; 0) ? a++ : a--; if (a &lt; 0)     if (a+b &gt; 0) c = a++;     else c = a--;                 </pre>	<pre> int a, b, c; c = (a == 0) ? b : a+1; if (a &lt; 0)     if (a == 0) c = b;     else c = a+1;                 </pre>
---	--

مثال 3 : اكتب شفرة لمقارنة قيمتي عددين صحيحين وطباعة مربع العدد الأصغر بينهما .

1. int x, y, z ;
2. Console.WriteLine("Enter x = "); x = Int32.Parse(Console.ReadLine());
3. Console.WriteLine("Enter y = "); y = Int32.Parse(Console.ReadLine());
4. z = (x < y) ? x \* x : y \* y ;
5. Console.WriteLine("z = {0}", z);

مثال 4 : اكتب شفرة لحساب القيمة المطلقة لعدد صحيح مدخل من لوحة المفاتيح .

1. int num;
2. Console.WriteLine("Enter number = "); num = Convert.ToInt32(Console.ReadLine());
3. int abs = num < 0 ? -num : num ;
4. Console.WriteLine("The ABS of number = {0} و abs) ;

$$ax^2 + bx + c = 0$$

مؤلف: حل معادلة من الدرجة الثانية

1. double a; String sa;
2. double b; String sb;
3. double c; String sc;
4. double delta, x1, x2, x2;
5. console.write("Input a = ");
6. sa = console.ReadLine(); a = Double.Parse(sa);
7. console.write("Input b = ");
8. sb = console.ReadLine(); b = Double.Parse(sb);
9. console.write("Input c = ");
10. sc = console.ReadLine(); c = Double.Parse(sc);
11. delta = b \* b - 4 \* a \* c;
12. if (delta < 0)
13. console.WriteLine("No Real Solution");
14. if (delta == 0)
15. {
16. console.WriteLine("one Solution");
17. x = -b / (2 \* a);
18. console.WriteLine("x = " + x);
19. }
20. if (delta > 0)
21. {
22. console.WriteLine("Two Solution");
23. x1 = (-b - Math.Sqrt(delta)) / (2 \* a);
24. x2 = (-b + Math.Sqrt(delta)) / (2 \* a);
25. console.WriteLine("x1 = " + x1 + "x2 = " + x2);
26. }

```
Switch (متحول/تعبير عربي) الشكل العام
{
  case value1:  ; تعليمة 1
    break;
  case value2:  ; تعليمة 2
    break;
  ...
  case valuen:  ; تعليمة n
    break;
  default:     ; تعليمة
    break;
}
```

1. تنفيذ تعليمة break في نهاية أي حالة من حالات switch في نقل التحكم مباشرة إلى خارج البنية switch ولابد من التنفيذ إلى حالات switch المتبقية .
2. في حال عدم تاري قيمة التعبير أي من القيم الموجودة بعد case فيتم تنفيذ العبارات الواقعة بعد التعليمة default وهي اختيارية (أي يمكن كتابة الكلمة default أو عدم كتابتها وذلك حسب الحاجة).

مثال 1

اكتب شيفرة بلغة C# تطلب من المستخدم إدخال عدد صحيح وإذا كانت قيمته (-1) يطبع عبارة "Welcome in C# in arabic" وإذا كانت قيمته (-2) يطبع عبارة "Welcome in C# in English" وإذا كانت غير ذلك يطبع "Welcome".

```

1. console.write("please Enter your selection -1, -2:");
2. int x = Int32.Parse(Console.ReadLine());
3. switch(x)
4. {
5.     case -1: Console.WriteLine("welcome in c# in arabic");
6.         break;
7.     case -2: Console.WriteLine("welcome in c# in English");
8.         break;
9.     default: Console.WriteLine("welcome");
10.        break;
11. }

```

خروج الشيفرة في حال إدخال القيمة -2 :-

```

please Enter your selection -1, -2: -2
welcome in c# in English

```

خروج الشيفرة في حال إدخال القيمة 5 :-

```

please Enter your selection -1, -2 : 5
welcome

```

مثال 2، اكتب شيفرة تقرأ عدداً صحيحاً (Value) وتطبع كلمة odd إذا كان العدد فردياً وكلمة even إذا كان العدد زوجياً باستخدام تعليمة Switch.

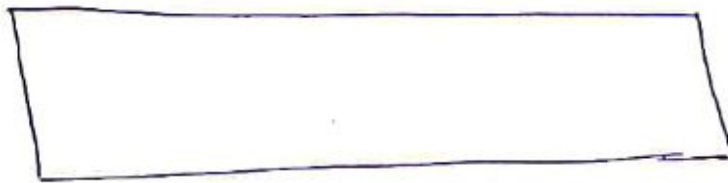
1. Int value و
2. Console.WriteLine("Enter integer number:");
3. value = Int.Parse(Console.ReadLine());
4. Switch(value % 2)
5. {
6. case 0 : Console.WriteLine("even");
7. break;
8. case 1 : Console.WriteLine("odd");
9. break;
10. }

```
Enter integer number : 34
even
```

```
Enter integer number : 21
odd
```

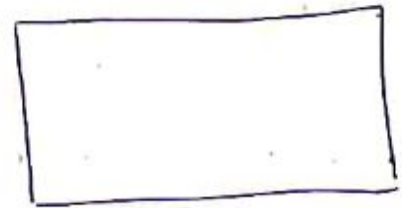
1 انطرح السيفرة الآتية :

```
string x;  
x = "Hello";  
if (x.Length >= 200)  
    Console.WriteLine("The Number is small");  
else  
    Console.WriteLine("The Number is big");
```



2 انطرح السيفرة الآتية :

```
int x;  
x = 5;  
if (x >= 5)  
    Console.WriteLine("x: {0}", x+2);
```



3 ما هو الناتج الذي يظهر على الشاشة عند تنفيذ جزء البرنامج التالي .

```
int i = 4, Var = 20;  
if (i <= 5) and || (Var > 20)  
{  
    i = i + 1;  
    if (i > 2) or || (Var > 20)  
        Var = 13;  
}  
Console.WriteLine(i + Var);
```

ما صرناج كلاضنا الشيفرات العاليه :

float a=2.5, b=7.5, c=5.5 ;

```
if (a > b) && (a > c) Console.WriteLine("max = " + a);  
else if (b > a) && (b > c) Console.WriteLine("max = " + b);  
else Console.WriteLine("max = " + c);
```

[4]

int x=5, y=3 ;

```
if (x > y) y = y - x;  
else x = x - y;
```

```
Console.WriteLine("x = " + x + " y = " + y);
```

[5]

int N=6, x=8 ;

```
if (N > 5) || (x < 8) n = n + 1;
```

```
if (x % n) < 5 x = x + 1;
```

```
Console.WriteLine(x + n);
```

[6]

تدريب 1: اكتب برنامج لايجاد العدد الاضغر بين ثلاثة اعداد مدخلة من لوحة المفاتيح وطباعة الناتج ، ثم تعديل البرنامج ليطلع العدد الاكبر.

تدريب 2: اكتب برنامج يدخل عدداً هجياً ويحدد فيما إذا كان زوجياً أو فردياً ، فإذا كان زوجياً يجمع مربعه ، وإذا كان فردياً يحسب مكعبه.

تدريب 3: اكتب برنامجاً يقرأ عددين حقيقيين مدخولين من لوحة المفاتيح.  
ويقرأ رمز العملية الحسابية المطلوبة تنفيذاً عن العددين الحقيقيين.

```
1. static void Main (string[] args)
2. { float x, y;
3.   console.write ("Enter x = ");
4.   x = single.parse (console.readline());
5.   console.write ("Enter y = ");
6.   y = single.parse (console.readline());
7.   char op;
8.   console.write ("please enter operation +, -, *, / :");
9.   op = char.parse (console.readline());
10.  switch (op)
11.  {
12.    case '+': console.writeline ("x+y={0}", x+y);
13.             break;
14.    case '-': console.writeline ("x-y={0}", x-y);
15.             break;
16.    case '*': console.writeline ("x*y={0}", x*y);
17.             break;
18.    case '/': {
19.               if (y == 0)
20.                 console.writeline ("sorry");
21.               else
22.                 console.writeline ("x/y={0}", x/y);
23.             } break;
24.    default: console.writeline ("please try again");
25.             break;
26.  }
27. }
28 }
```

```
1. int m; string sm;  
2. int mDays = 0;  
3. // Read m  
4. Console.WriteLine("Input m:");  
5. sm = Console.ReadLine(); m = Int.Parse(sm);  
6. switch (m)  
7. {  
8.     case 1 :  
9.         case 3 :  
10.        case 5 :  
11.        case 7 :  
12.        case 8 :  
13.        case 10 :  
14.        case 12 :  
15.            mDays = 31 ;  
16.            break ;  
17.        case 4 :  
18.        case 6 :  
19.        case 9 :  
20.        case 11 :  
21.            mDays = 30 ;  
22.            break ;  
23.        case 2 :  
24.            mDays = 28 ;  
25.            break ;  
26.    }  
27. Console.WriteLine("Month " + m + " has " + mDays + " days");
```

Input m: 1  
Month 1 has 31 days

Input m: 4  
Month 4 has 30 days

Input m: 2  
Month 2 has 28 days

## بنى التكرار والحلقات

- الحلقات نوعين: - الحلقة ذات العداد .  
- الحلقة الشرطية .

• تستخدم الحلقات لتكرار مجموعة محددة من التعليمات لعدد محدد من المرات .

### □ الحلقة ذات العداد for

الشكل العام  
for (type var = Value ; Condition ; Increasing  
or  
decreasing )  
{  
Statements ;  
}

- ملاحظات
- 1- تقوم عبارة for بتنفيذ التعليمات Statements وفي المرة الثانية كترداد أو تنقص قيمة العداد بمقدار الزيادة أو النقصان، وهكذا حتى يصل العداد إلى القيمة النهائية ويتم إنشاء عمليات التكرار والمزج من الحلقة For .
  - 2- إذا كانت الحلقة ستكرر تعليمة واحدة فقط فلا داعي لاستخدام الأقواس {} .
  - 3- تقوم الحلقة For بتنفيذ تعليمات جسم الحلقة حتى تصبح قيمة العداد خارج مجال قيم العداد .

```
for (int i = 1 ; i <= 5 ; i++)  
Console.WriteLine ( i + "Hi" );
```

وإذا أردنا كتابة المثال باستخدام for المتناقص

```
for (int i = 5 ; i >= 1 ; i--)  
Console.WriteLine ( i + "Hi" );
```

مثال 1:

1Hi
2Hi
3Hi
4Hi
5Hi

5Hi
4Hi
3Hi
2Hi
1Hi

مثال 2، اكتب شيفرة اختيار أكبر عدد من مجموعة أعداد مدخلة من لوحة المفاتيح [أعداد موجبة] المؤلفة من n عدد صحيح.

```
1. int i, n, x, max=0;
2. Console.WriteLine("n=");
3. n = int.Parse(Console.ReadLine());
4. for(i=1; i<=n; i++)
5. {
6.     Console.WriteLine("x=");
7.     x = int.Parse(Console.ReadLine());
8.     if(x > max)
9.         max = x;
10. }
11. Console.WriteLine("max={0}", max);
```

```
n = 3
x = 20
x = 25
x = 15
max = 25
```

مثال 3، اكتب شيفرة تقوم بحساب قواسم عدد صحيح باستخدام الكلفة for.

```
1. int x, i;
2. Console.WriteLine("x=");
3. x = int.Parse(Console.ReadLine());
4. for(i=1; i<=x; i++)
5. {
6.     if((x%i) == 0)
7.         Console.WriteLine(i);
8. }
```

```
x = 12
1
2
3
4
6
```

مثال 4، حساب مجموع الأعداد من 1 حتى m تعطى m تعطى بالدفء.

1. int m, string sm;
2. int i, s;
3. Console.WriteLine("Input m = ");
4. sm = Console.ReadLine();
5. m = Int32.Parse(sm);
6. s = 0;
7. for(i=1; i<=m; i++)
8. {
9. s = s + i;
10. }
11. Console.WriteLine("1+2+.....+ m + " = " + s);

```
Input m = 4
1+2+.....+4 = 10
```

1.  $N! = 1 \times 2 \times 3 \dots \times N$  مثال 5: حساب N! تعطى N تعطى كدفء
2. int N, s, i;
3. Console.WriteLine("Input N:");
4. N = Int32.Parse(Console.ReadLine());
5. s = 1;
6. for(i=1; i<=N; i++)
7. s = s \* i;
8. Console.WriteLine(N + "! = " + s);

```
Input N: 5
5! = 120
```

```
while (condition)
{
    statements;
}
```

الشكل العام

• عمل الكلفة : يتم فحص الشرط أولاً ، إذا كان الشرط True يدخل الكلفة while وينفذ

التعليمات او عند نهاية الكلفة يعاود للتحقق من جديد من الشرط وطالما

الشرط True يستمر بتكرار تعليمات الكلفة .

وعندما يصبح False يتوقف عن التكرار وينتقل إلى التعليمة التي تلي ضيق while .

مثال 1 :

```
int i = 0;
int x = 0;
while (i <= 10)
{
    x = x + 10;
    i++;
}
```

مثال 2: اكتب شيفرة تقوم بجمع أعداد مدخلة من قبل المستخدم وحساب المتوسط الحسابي لها وطباعة النتائج.

1. `int i, n, sum = 0; float avg;`
2. `console.write("n = ");`
3. `int n = int.Parse(console.ReadLine());`
4. `i = 1; // The initial value of i`
5. `while(i <= n) // The condition`
6. `{`
7. `console.write("x = ");`
8. `x = int.Parse(console.ReadLine());`
9. `sum = sum + x;`
10. `i += 1; // increase i`
11. `}`
12. `avg = (float) sum / n;`
13. `console.WriteLine("sum = {0}", sum);`
14. `console.WriteLine("avg = {0}", avg);`

`n = 4`  
`x = 2`  
`x = 3`  
`x = 5`  
`x = 1`  
`sum = 11`  
`avg = 2.75`

مثال 3 ، اكتب شيفرة لحساب القاسم المشترك الأعظم ومنه خوارزمية إقليدس

خوارزمية إقليدس : نطرح الأصغر من الأكبر ونضع الناتج مكان الأكبر، نكرر ذلك حتى يتساوى العددين

مثال 45 و 75

A	B
75	45
30	45
30	15
15	15

القاسم المشترك الأعظم

1. int a; String sa;
2. int b; String sb;
3. int g;
4. // Read a, b
5. console.write("a = ");
6. sa = Console.ReadLine(); a = Int.Parse(sa);
7. console.write("b = ");
8. sb = Console.ReadLine(); b = Int.Parse(sb);
9. while (a != b)
10. {
11. if (a > b)
12. a = a - b;
13. else
14. b = b - a;
15. }
16. g = a; <sup>a</sup>/<sub>b</sub>
17. console.WriteLine("GCD = " + g);

a = 75
b = 45
GCD = 15

مثال 4 : اكتب شفرة لطباعة جدول الضرب المدرسي لعدد k من المجال [2..10] حيث نقرأ قيمة k من الرطل .

1. int k, i, String sk;
2. Console.WriteLine("Input k in [2..10] = ");
3. sk = Console.ReadLine(); k = Int32.Parse(sk);
4. i = 1;
5. while (i <= 10)
6. {
7. Console.WriteLine(k + "\*" + i + "=" + k \* i);
8. i = i + 1;
9. }

```
Input k in [2..10] = 3
3 * 1 = 3
3 * 2 = 6
3 * 3 = 9
3 * 4 = 12
3 * 5 = 15
3 * 6 = 18
3 * 7 = 21
3 * 8 = 24
3 * 9 = 27
3 * 10 = 30
```

مثال 5 : شفرة لطباعة كل جدول الضرب للأعداد من 1 حتى 10

1. int k, i;
2. k = 1;
3. while (k <= 10)
4. {
5. i = 1;
6. while (i <= 10)
7. {
8. Console.WriteLine(k + "x" + i + "=" + k \* i);
9. i = i + 1;
10. }
11. Console.WriteLine(".....");
12. k = k + 1;
13. }

تمرين : عدّل البرنامج السابق للوصول إلى جدول الضرب باستخدام صلتات For

مثال 6: اكتب صنف Average بلفئة #C ياعد في حاسب المتوسط الكما في عشرة  
أرقام صحيحة مجرى طلبا من البرنامج وإدخالها من قبل المستخدم.

```
1. using System;
2. namespace Average
3. {
4.     class Average1
5.     {
6.         static void Main(string[] args)
7.         {
8.             int total // مجموع الدرجات
9.             gradeCounter // عدد الدرجات
10.            gradeValue // قيمة الدرجة
11.            average // المتوسط
12.            total = 0;
13.            gradeCounter = 1;
14.            while (gradeCounter <= 10)
15.            {
16.                Console.WriteLine("Enter integer grade:");
17.                gradeValue = Int32.Parse(Console.ReadLine());
18.                total = total + gradeValue;
19.                gradeCounter = gradeCounter + 1;
20.            }
21.            average = total / 10;
22.            Console.WriteLine("Average in class average is " + average);
23.        }
24.    }
25. }
```

مثال 7، حساب المتوسط الحسابي لعدد من الأرقام الصحيحة يدخل المستخدم ويحتمل التوقف  
طلب الأعداد عند إدخال المستخدم لقيمة -1.

1. int total, gradeCounter, gradeValue;
2. double average;
3. total = 0;
4. gradeCounter = 0;
5. Console.WriteLine("Enter Integer grade, -1 to out: ");
6. gradeValue = Int32.Parse(Console.ReadLine());
7. while (gradeValue != -1)
8. {
9. total = total + gradeValue;
10. gradeCounter += 1;
11. Console.WriteLine("Enter Integer grade, -1 to out: ");
12. gradeValue = Int32.Parse(Console.ReadLine());
13. }
14. if (gradeCounter != 0)
15. {
16. average = (double) total / gradeCounter;
17. Console.WriteLine("In class average is: " + average);
18. }
19. else
20. Console.WriteLine("No grades were entered.");

مثال 8: بفرض لديك 10 طلاب ، اكتب برنامج يسقرفض رقم الطالب من 1 إلى 10 ثم تحآ  
يحيى يقوم المستخدم من أجل كل طالب بإدخال رقم 1 في حال كان الطالب ناجح  
و يقوم بإدخال رقم 2 في حال كان الطالب راسب  
يحيى يعطيه البرنامج عدد الناجحين وعدد الراسبين والنسبة المئوية للنجاح.

```
1. int passes=0, failures=0, student=1, result;  
2. while (student <=10)  
3. {  
4.     console.write ("student "+student + " result (1=Pass, 2=Fail) = ");  
5.     result = int32.Parse (console.ReadLine());  
6.     if (result == 1)  
7.         passes = passes+1 ;  
8.     else  
9.         failures = failures+1 ;  
10.    student = student +1 ;  
11. }  
12. console.WriteLine ();  
13. console.WriteLine ("passed : " + passes);  
14. console.WriteLine ("failures : " + failures );  
15. console.WriteLine ("Raise Percentage \n" + passes * 10);
```

مثال 9 : جرب البرنامج التالي وأب. ملاحظاته في عمليات الزيادة بقيمة 1.

```
1. static void Main(string[] args)
2. {
3.     int c;
4.     c = 5;
5.     Console.WriteLine(c);
6.     Console.WriteLine(c++);
7.     Console.WriteLine(c);
8.     Console.WriteLine(c);
9.     c = 5;
10.    Console.WriteLine(c);
11.    Console.WriteLine(++c);
12.    Console.WriteLine(c);
13. }
```

مثال 10 : اكتب برنامج يحاسب مجموع الأعداد الزوجية المتصورة بين 0 و 100.

```
1. int sum = 0;
2. int number = 2;
3. while (number <= 100)
4. {
5.     sum += number;
6.     number += 2;
7. }
8. Console.WriteLine("The sum is " + sum + " sum even Integers from 2 to 100");
```

The sum is 2250 sum even Integers from 2 to 100

```

1. static void Main (string [] args)
2. {
3. // testing the conditional AND operator (&&)
4. console.WriteLine ("Conditional AND (&&) " +
5.     " \n false && false : " + (false && false) +
6.     " \n false && true : " + (false && true) +
7.     " \n true && false : " + (true && false) +
8.     " \n true && true : " + (true && true) );
9. // testing the conditional OR operator (||)
10. console.WriteLine ("Conditional OR (||) " +
11.     " \n false || false : " + (false || false) +
12.     " \n false || true : " + (false || true) +
13.     " \n true || false : " + (true || false) +
14.     " \n true || true : " + (true || true) );
15. // testing the conditional exclusive OR operator (^)
16. console.WriteLine ("Conditional exclusive OR (^) " +
17.     " \n false ^ false : " + (false ^ false) +
18.     " \n false ^ true : " + (false ^ true) +
19.     " \n true ^ false : " + (true ^ false) +
20.     " \n true ^ true : " + (true ^ true) );
21. // testing the conditional Not operator (!)
22. console.WriteLine ("Logical Not (!) " +
23.     " \n !false : " + (!false) +
24.     " \n !true : " + (!true) );
    
```

نتيجة &

1

### 3 الكلفة الشرطية do... while (كرملة واحدة):

```
do  
{  
    statements;  
}  
while (condition);
```

الشكل العام:

• يتم تنفيذ التعليمات أولاً ثم فحص الشرط أو شرط التوقف عن تكرار تنفيذ هذه التعليمات هو اختبار منطقي يُجرى عند نهاية هذه التعليمات أو في نهاية جسم الكلفة.

مثال 1: اكتب شيفرة لجمع الأعداد الموصية والمهذلة من لوحة المفاتيح.

```
1. int n, sum ;  
2. sum = 0 ;  
3. do {  
4.     n = Int32.Parse ( Console.ReadLine ( ) );  
5.     if ( n > 0 )  
6.         sum = sum + n ;  
7. }  
8. while ( n >= 0 );  
9. Console.WriteLine ( "sum = {0}" و sum );
```

حساب المجموع لمجموعة من الأعداد الصحيحة - بجزء ادخال البرنامج (تتبعي الادخال بالقيمة 999).

```
1. int n=0 ; String sn ; int sum=0 ;
2. do
3. {
4.     sum = sum + n ;
5.     console.write ("Input n: ") ;
6.     sn = console.ReadLine() ; n = Int32.Parse (sn) ;
7. }
8. while (n != 999) ;
9. console.WriteLine ("Sum = " + sum) ;
```

```
Input n : 5
Input n : 8
Input n : 2
Input n : 999
Sum = 15
```

```
Input n : 999
Sum = 0
```

تستخدم تعليمات التحكم بالكلمات من أجل تجاوز تنفيذ بعض دوراتها أو من أجل إيقافها والانتقال إلى التعليمات الآتية لتنفيذ الكلفة وهذه التعليمات هي:

1] تعليمة الإيقاف break :

تستخدم داخل كتلة تعليمات معينة من أجل إيقاف تنفيذها والكروج إلى ما بعد نهايتها كتلة التعليمات أو غالباً تستخدم التعليمة break ضمن بنية اختيارية أو شرطية.

مثال :

```
int i=0, n=13;
```

```
while (i <= n)
```

```
{ if ((i%4) == 0) && (i > 4)
```

```
break;
```

```
console.WriteLine("i = " + i);
```

```
i = i + 1;
```

```
}
```

i = 0
i = 1
i = 2
i = 3
i = 4
i = 5
i = 6
i = 7

نفس المثال السابق باستخدام الكلفة for

```
for (int i=0, i<=13; i++)
```

```
{ if ((i%4) == 0) && (i > 4)
```

```
break;
```

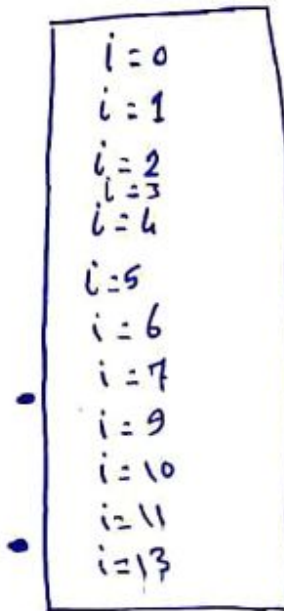
```
console.WriteLine("i = " + i);
```

```
}
```

توضح هذه التعليمة في الكلفة وتستخدم ضمن بنى شرطية أو اختيارية ، إذا تحققت الشرط في البنية الشرطية يتم تجاوز تنفيذ بقية تعليمة الكلفة فلال الدورة الكلية والانتقال إلى بداية الكلفة لتنفيذ الدورة التالية من الكلفة .

مثاله :

```
for (int i=0 ; i<=13 ; i++)  
{  
    if ((i%4)==0) && (i>4)  
        continue ;  
    console.WriteLine("i = " + i) ;  
}
```



```
① int x;  
for(x=0; x<=10; x++) {  
    if(x==1)  
        console.write("x={0}", x);
```



```
② int x;  
for(x=5; x>1, x--)  
{  
    console.write(" {0}", x);
```



```
③ int x=2;  
do  
{  
    x=x+2;  
    console.write(x);  
} while(x<=10);
```



[2] المكتبة برنامج بلغة #C للايجاد وطباعة عدد الأعداد الفردية وعدد الأعداد الزوجية ضمن المجال 1 وحتى العدد n باستخدام اللفظة for.

1. static void Main (String[] args)
2. {
3. int sum\_odd = 0, sum\_even = 0;
4. console.WriteLine ("n = ");
5. int n = Int32.Parse (console.ReadLine());
6. for (int i = 1; i <= n; i++)
7. {
8. if (i % 2 != 0)
9. sum\_odd = sum\_odd + 1;
10. else
11. sum\_even = sum\_even + 1;
12. }
13. console.WriteLine ("sum\_odd = {0}", sum\_odd);
14. console.WriteLine ("sum\_even = {0}", sum\_even);

المرجع التنفيذي يفرض n = 25

n = 25
sum_odd = 13
sum_even = 12

[3] عدل على البرنامج السابق لإيجاد مكعب العدد الفردي من 1 إلى n ، حيث n عدد صحيح  
مدخل من لوحة المفاتيح.

```
static void Main(string[] args)
{
    Console.WriteLine("n = ");
    int n = Int32.Parse(Console.ReadLine());
    for (int i = 1; i <= n; i += 2) // العدد يزداد بقدار 2
    {
        Console.WriteLine("Cubic_odd = {0}", i * i * i);
    }
}
```

n = 7
Cubic_odd = 1
Cubic_odd = 27
Cubic_odd = 125
Cubic_odd = 343

[4] تدريج : كتابة برنامج لإيجاد مربعات الأعداد الزوجية من 2 حتى 80 .

5] اكتب برنامج لقرأ أعداد صحيحة وعقب مجموعها وتجاه إيقاف تنفيذ الحلقة باستخدام تعليمة break عند ادخال قيمة صفرية.

1. Static void Main(String[] args)
2. {
3.     int sum = 0;
4.     console.WriteLine("If you want to exit, Enter 0");
5.     for ( ; ) // حلقة for لانهائة
6.     {
7.         console.WriteLine("x = ");
8.         int x = Int32.Parse(console.ReadLine());
9.         sum += x;
10.         if (x == 0)
11.             break;
12.     }
13.     console.WriteLine("Sum = " + sum);
14. }

```
If you want to exit, Enter 0
x = 4
x = 5
x = 7
x = 0
Sum = 16
```

أنماط المخطات المركبة

array الجداول [2]

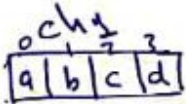
سلاسل المخابرات

سلاسل المخابرات: صف من المكتبة system

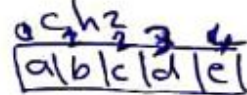
- التصنع عن سلة مخابرات

String ch1;

← ch1 = "abcd" ;

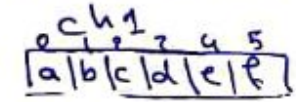


String ch2 = "abcde" ; ⇒

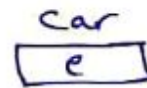


- يمكن الوصول إلى أحد مخابرات السلة باستخدام العملية [] :

String ch1 = "abcdef" ; ⇒



char car = ch1[4] ; ⇒

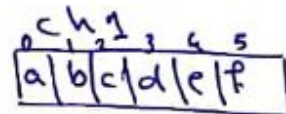


\* لا يمكن تعديل مخراف باستخدام العملية [] .

ch1[7] = car ; // error

التعديل: المخر Insert :

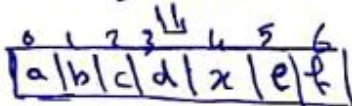
String ch1 = "abcdef" ; ⇒



String ch2 ;

ch2 = ch1.Insert(4, "x") ;

ch2



② الدمج باستخدام عملية +

```
String str1, str2, str3;  
str1 = "bon";  
str2 = "jour";  
str3 = str1 + str2;  
    ↓  
str3 bonjour
```

③ الكسول على طول السلسلة المرفقة Length

```
String str4 = "abcdef";  
int len;  
len = str4.length;  
    ↓  
len 6
```

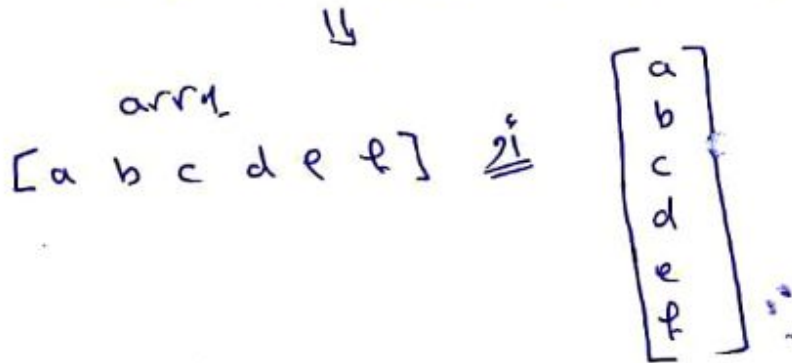
④ التعليق: الكسول على موقع سلسلة جزئية داخل سلسلة محارف indexOf

\* تعيد عدد صحيح لترقم موقع أول ظهور للسلسلة الجزئية.

```
String str5 = "abcdef", ssch = "cde";  
int ord;  
ord = str5.indexOf(ssch);  
    ↓  
ord 2
```

⑤ التحويل، تحويل سلسلة محارف إلى جدول محارف ToCharArray

```
string str = "abcdef";  
char[] arr;  
arr = str.ToCharArray();  
arr = "abcdef".ToCharArray();
```



⑥ التحويل: الاسناد والمقارنة

```
string s1, s2, s3, s4, ch;  
ch = "abcdef";  
s1 = ch;  
s2 = "abcdef";  
s3 = new string("abcdef".ToCharArray());  
s4 = new string(s3.ToCharArray());  
console.WriteLine("s1 = " + s1);  
console.WriteLine("s2 = " + s2);  
console.WriteLine("s3 = " + s3);  
console.WriteLine("s4 = " + s4);  
console.WriteLine("ch = " + ch);
```

```
if (s2 == ch)
```

```
    console.WriteLine("s2 == ch");
```

```
else
```

```
    console.WriteLine("s2 <> ch");
```

```
if (s3 == s4)
```

```
    console.WriteLine("s3 == s4");
```

```
else
```

```
    console.WriteLine("s3 <> s4");
```

```
if (s3.Equals(ch))
```

```
    console.WriteLine("s3 == ch");
```

```
else
```

```
    console.WriteLine("s3 <> ch");
```

[2] أنماط المعطيات المركبة، الجداول والمصفوفات : [جميع عناصر المصفوفة لها نفس نوع المعطيات]

```
int [] table1;  
float [] A;  
char [] table2;  
string [] tablestr;
```

- تعريف مصفوفة من دون تحديد حجم

- تعريف مصفوفة مع تحديد حجم :

```
int [] table1 = new int [5];  
char [] table2 = new char [8];
```

- تعريف مصفوفة مع إعطائها قيماً ابتدائية مباشرة

```
int [] table1 = {17, -9, 4, 3, 57};  
char [] table2 = {'a', 'b', 'c', 'd'};
```

مثال

```
int [] table1 = new int [5];  
table1[0] = -458;  
table1[4] = 5891;  
table1[5] = 72; // error
```

فتح ادخال وإخراج قيم عناصر المصفوفة من لوحة المفاتيح باستخدام الحلقات :

مثال 1: اكتب شيفرة كساب مجموع عناصر مصفوفة مدخلة أحادية البعد (شعاع):

```
int [] a = new int [5];  
int sum = 0;  
for (int i=0; i < a.Length; i++) // For reading the array  
{  
    Console.WriteLine("Enter a element");  
    int x = Int32.Parse(Console.ReadLine());  
    a[i] = x;  
}  
for (int i=0; i < a.Length-1; i++) // The sum of elements  
{  
    sum = sum + a[i];  
}  
Console.WriteLine("The sum of elements: {0}", sum);  
for (int i=0; i < a.Length; i++) // For writing the array  
{  
    Console.WriteLine("a[" + i + "] = " + a[i]);  
}
```

```
int [] table1 = new int [5]
```

مثال 2

```
for (int i=0; i<=table1.length-1; i++)
```

```
table1[i] = 3*i - 1;
```

```
for (int i=0; i<=table1.length-1; i++)
```

```
console.WriteLine (table1[i]);
```

-1
2
5
8
11

```
char [] t2 = new char [7];
```

```
t2[0] = '?';
```

```
t2[1] = 'a';
```

```
t2[4] = '#';
```

مثال 3

```
for (int i=0; i<=t2.length-1; i++)
```

```
t2[i] = (char) ('a' + i);
```

```
for (int i=0; i<7; i++)
```

```
console.WriteLine ( ('a' + i) + "\t" + t2[i] );
```

97	a
98	b
99	c
100	d
101	e
102	f
103	g

```
int m; String sm;  
int [] Mdays = { 31, 28, 31, 30, 31, 30, 31, 31, 30, 31, 30, 31 };  
console.write ("Input m = ");  
sm = console.ReadLine(); m = Int32.Parse (sm);  
console.WriteLine ("Month " + m + " has " + Mdays[m-1] + " days");
```

```
Input m = 1  
month 1 has 31 days
```

مثال 5: قراءة في جدول وحاجب القيمة العظمى والعظمى والمتوسط الحسابي  
الذات: عدد عناصر الجدول، وفيه مثلثات الطلاب.

```
int N; string sN; int [] T; int i, minT, maxT;  
float sum;  
console.write ("Input Total Student Number: ");  
sN = console.ReadLine(); N = Int32.Parse (sN);  
T = new int [N];  
for (i = 0; i <= N-1; i++)  
{  
    console.write ("Input score: ");  
    sN = console.ReadLine();  
    T[i] = Int.Parse (sN);  
}
```

```
sum = minT = maxT = T[0];
```

```
for (i=1; i <= T.length-1, i++)  
{
```

```
    sum = sum + T[i];  
    if (T[i] < minT)  
        minT = T[i];
```

```
    if (T[i] > maxT)  
        maxT = T[i];
```

```
}
```

```
console.WriteLine();
```

```
console.WriteLine("Min=" + minT);
```

```
console.WriteLine("Max=" + maxT);
```

```
console.WriteLine("Average=" + (sum / N));
```

```
Input Total student Number: 5
```

```
Input score: 80
```

```
Input score: 65
```

```
Input score: 69
```

```
Input score: 75
```

```
Input score: 81
```

```
Min = 65
```

```
Max = 81
```

```
Average = 74
```

الشكل العام للتصريح بمصفوفة ثنائية :

Type [, ] name = new Type [Name of Row, Name of Column];

مثال:

مصنوفة من نظرين و 3 أعمدة //  $int [, ] x = new int [2, 3]$

$x[0,0] = 100;$

$x[0,1] = 200;$

$x[0,2] = 300;$

$x[1,0] = 400;$

$x[1,1] = 500;$

$x[1,2] = 600;$

يمكن اختصار الشطر البرمجي السابق والتصريح عن المصفوفة بإعطاء القيم الابتدائية لها صراحة

$int [, ] x = new int [2, 3] \{ \{ 100, 200, 300 \}, \{ 400, 500, 600 \} \};$

مثال 2: ادخال وافراج في عناصر مصفوفة ثنائية بن نظرين وثلاثة أعمدة.

$int [, ] x;$

$int [, ] x = new int [2, 3];$

for (i=0; i<2; i++) // For reading the elements of array

{ for (j=0; j<3; j++)

$x[i,j] = Int32.Parse(Console.ReadLine());$

}

```

for(i=0; i<2; ++i) // For writing the elements of array
for(j=0; j<3; ++j)
{
    console.write("x[", i, "],");
    console.write(" ", j, "]=");
    console.write(" ", x[i, j], "\t");
}

```

1			
2			
3			
4			
5			
6			
$x[0,0]=1$	$x[0,1]=2$	$x[0,2]=3$	$x[1,0]=4$
$x[1,1]=5$	$x[1,2]=6$		

تمرين 1: كتابة برنامج لطباعة عناصر القطر الرئيسي لمصفوفة أعداد صحيحة ثنائية البعد (3x3) مع صياغة نتائج جميع عناصر القطر الرئيسي.

```

int [,] arr = new int [3,3];
int i, j, sum=0;
for(i=0; i<3; i++) // For reading the array
{
    for(j=0; j<3; j++)
        arr[i, j] = Int32.Parse(Console.ReadLine());
}

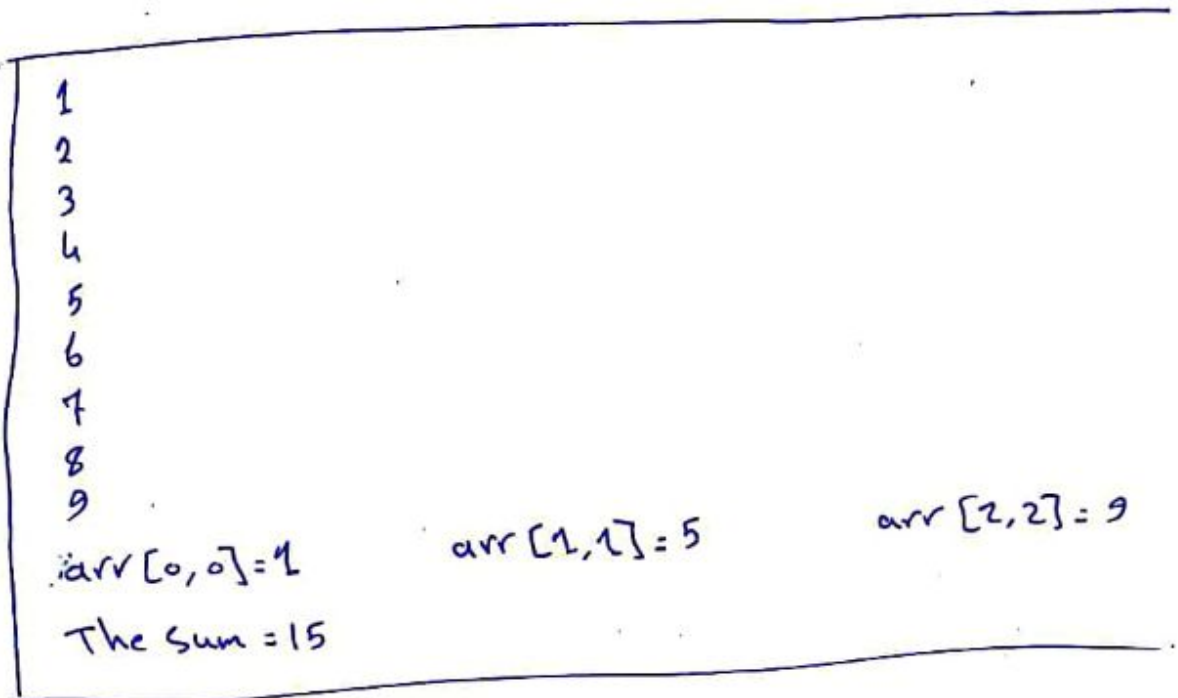
```

Milestone 4  
كان ؟

```
for (i = 0; i < 3, i++)  
{  
  console.write ("arr [ {0} , i );  
  console.write (", {0} ", i);  
  console.write (" ] = {0} \t , arr [i, i] );  
}
```

```
for (i = 0; i < 3, i++)  
  for (j = 0; j < 3, j++)  
  {  
    if (i == j)  
      sum = sum + arr [i, j];  
  }
```

```
console.writeLine (" \n The sum = " + sum);
```



تمرين 2: اكتب برنامج يحسب ناتج جمع صفحتين ثنائيتين البعد (2x2) وحفظ الناتج في صفوة ثالثة (2x2) وطباعة الناتج.



تمرين 4: اكمل البرنامج التالي الذي يساعدك في البحث عن رقم elt ضمن جدول،  
وعند إعطاء رسالة تؤكد وجود أو عدم وجود elt في الجدول مع إعطاء ترتيبه  
في حال وجوده.

```
int[] table = {12, -5, 7, 8, -6, 6, 4, 78};
```

```
int elt = 4, i;
```

```
for (i = 0; i < 8; i++)  
    if (elt == table[i]) break;  
  
if (i == 8)  
    console.WriteLine("Value: " + elt + " not Found");  
else  
    console.WriteLine("Value: " + elt + " order: " + i);
```

Value : 4 order 6

يهدف أسلوبه برمجية البرائيات والروال إلى الأمور الآتية:

- ① تقليل حجم البرنامج إذ يكفي كتابة التابع أو الإجراء لمرة واحدة فقط، واستعادته عدة مرات.
- ② تسهيل عملية البرمجة بتقسيم البرنامج إلى كتل برمجية يمكن المبرمج تطوير أي واحدة منها.
- ③ تنظيم البرنامج بتقسيمه إلى كتل برمجية.

### التوابع Functions

هي مجموعة من التعليمات أو كتلة من التعليمات يتم استدعاؤها عند الحاجة لها،

- وتصنف قسمة واحدة وتصنفه التوابع إلى:
- 1- التوابع المضمرة (المجازة).
  - 2- التوابع المعروفة من قبل المستخدم.

\* التوابع المضمرة موجودة في مكتبات خاصة. ونم استعاء التابع بذكر اسمه فقط مع وسطائه ضمن البرنامج أو مشرد المكتبات التي تحتوي التوابع المضمرة هي مكتبة Math وتحتوي على التوابع الآتية:

تستخدم التوابع بالشكل العام الآتي:  $\text{LibraryName.FunctionName(variable)}$

الوظيفة	صيغة الدالة	الوظيفة	صيغة الدالة
جيب الزاوية	$\text{Math.Sin}(x)$	القيمة المطلقة لعدد	$\text{Math.Abs}(x)$
تجيب الزاوية	$\text{Math.Cos}(x)$	تقريب لأقرب عدد صحيح	$\text{Math.Round}(x)$
ظل الزاوية	$\text{Math.Tan}(x)$	تقريب إلى العدد الصحيح السابق	$\text{Math.Floor}(x)$
العدد النيري مرفوع لقوة $x$	$\text{Math.EXP}(x)$	تقريب إلى العدد الصحيح التالي	$\text{Math.Ceiling}(x)$
اللوغاريتم الطبيعي للعدد $x$	$\text{Math.Log}(x)$	الرفع لقوة $x^y$	$\text{Math.Pow}(x, y)$
اللوغاريتم العشري للعدد $x$	$\text{Math.Log}_{10}(x)$	الجذر التربيعي لعدد	$\text{Math.Sqrt}(x)$

هي التواب التي يقوم المبرمج بكتابتها ضمن البرنامج وتعيد قيمة واحدة من أحد أنواع المعطيات  
و يتم استدعاؤها من قبل البرنامج الرئيسي او كتعبير شيفرة التاج في قسم منفصل عن البرنامج  
الرئيسي.

- يمكن أن يحتوي البرنامج على عدة تواب حسب حاجتنا لها.

- ويتم استدعاء التاج بإضافة اسمه إلى متحول أو ضمن تعبير صائ أو منطقي أو واحدة  
من تعاليم لغة #C والشكل العام للتصريح عن التاج هو:

```

Static Type_of_Function Name_of_Function (Parameters)
{
    Function code
    ...
    return Value;
}

```

مثال: اكتب برنامجاً يجمع عددين باستخدام أسلوب البرمجة التواب

1. class Program

2. { static void Main (string[] args)

3. { console.write ("num1 = ");

4. int num1 = Int32.parse (console.ReadLine());

5. console.write ("num2 = ");

6. int num2 = Int32.parse (console.ReadLine());

7. int s = sum (num1, num2); // call function

8. console.WriteLine ("sum = {0}", s);

9. } static int sum (int x, int y) // التصريح عن التاج

10. int z;

11. z = x + y;

12. return z;

13. }

14. }

num1 = 4
num2 = 5
sum = 9

الاجراء: هو كتلة برمجية مؤلفة من مجموعة تعليمات تُنفذ عند استدعائها، وتقوم بتوليفة محددة كالتوليفة على الشاشة أو صواب فصح أو واحدة أو عدة قيم والشكل العام للاجراء هو:

```
static void Name_of_Procedure (Parameters)
{
    Procedure code.
}
```

يمكن الاستغناء عنها عند الرغبة بإرجاع قيمة

مثال: اكتب برنامج يقوم بحساب تكعب عدة أعداد صحيحة وذلك من لوحة المفاتيح ومن إيقاف التنفيذ عندما يبرطل المستخدم القيمة 0 باستخدام أسلوب برمجية الاجراءات.

1. class program

```
2. { static void Main (string[] args)
3. {
4.     int x;
5.     for ( ; ; )
6.     { console.WriteLine ("If you want to exit press 0");
7.       console.write ("x = ");
8.       x = Convert.ToInt32 (console.ReadLine());
9.       if (x == 0)
10.      { console.WriteLine ("Sorry I want to exit");
11.        break;
12.      }
13.      } else print (x);
14.    }
15.  }
16.  static void print (int x)
17.  {
18.      console.WriteLine ("The cubic of x = {0}", x * x * x);
19.  }
20.  }
21.  }
22. }
```

تكرر برصا مكررات

• يتم استدعاء الإجراءات ضمن ال Main بتعليمات مستقلة .

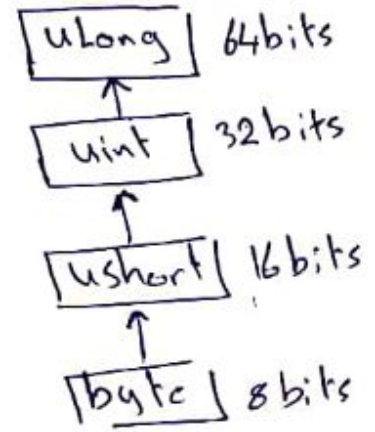
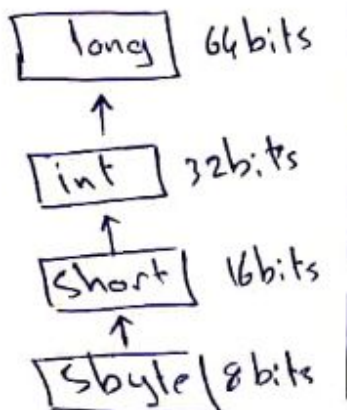
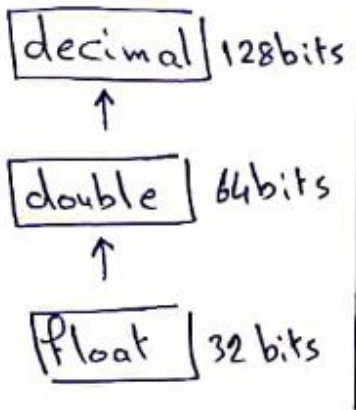
مثال 1

```
1. class APP
2. {
3.     static void Main (string [] args)
4.     {
5.         int [] table = {12, -5, 7, 8, -6, 6, 4, 78, 2};
6.         long elt = 4 ; int i ;
7.         for (i=0 ; i<8 ; i++)
8.             if (elt == table [i]) break ;
9.         Display (i, elt) ;
10.    }
11.    static void Display (int ord, long val)
12.    {
13.        if (ord == 8)
14.            console.WriteLine ("value : " + val + " Not Found.");
15.        else
16.            console.WriteLine ("value : " + val + " order : " + ord);
17.    }
18. }
```

تمرين عدل المثال السابق بحيث تقوم المستخدم بإدخال الرقم الذي تبحث عنه في الكود  
وأجعل البرنامج ليكرر حتى تقوم المستخدم بإدخال قيمة 0 فيوقف كود البرنامج .

- يمكن للعقول ذات النمط صاغر الحجم الأكبر أن تستوعب قيمة تنتمي إلى نمط قجانه معه و ذر حجم أصغر.

- يظهر هذا الكلام على تمثيل المعاملات.



• تمثيل المعاملات - تمثيل القيمة:

عند استعاء طريقة تمتلك معامل ممر بالقيمة (من أجل متحول ما) يقوم ببناء نسخة من المتحول (نسخة من قيمته) وتحويلها إلى الطريقة بحيث لا يؤثر أي تعديل على النسخة الممررة على المتحول الأصلي (نسخة تلقائياً تكون تمثيل المعاملات للطرز تمثيلاً بالقيمة).

```

1. static void Main (String [] args)
2. {
3.     int [] table = {12, -5, 7, 8, -6, 6, 4, 78, 2};
4.     int res = 0;
5.     short i;
6.     for (i=0; i<=8; i++)
7.     {
8.         res = compute (i, res, table[i]);
9.     }
10.    console.WriteLine ("Result : " + res);
11. }
12. static int compute (short s; int r; int val)
13. {
14.     int k = r + s * val;
15.     return k;
16. }

```

```

Result: 0
Result: -5
Result: 9
Result: 33
Result: 9
Result: 39
Result: 63
Result: 609
Result: 625

```

تمرين: [تطبيق على فكرة التمرير بالقيمة]:

ما هي نتيجة k التي تظهر نتيجة تنفيذ البرنامج التالي؟

1. static void Main (String [] args)
2. {
3. int [] table = {12, -5, 7, 8, -6, 6, 4, 78, 2};
4. int k = 0;
5. short i;
6. for (i:0; i < 8; i++)
7. Increment (k);
8. console.WriteLine ("Result: " + k);
9. }
10. static void Increment (int val).
11. {
12. val = val + 1;
13. }

Result : 0

## • تمرير المعاملات - تمرير العنوان : أر بالمرجع أو بالمرش :

- يكون أسلوب تمرير عناوين المتولات بوضع الكلمة المفتاحية ref :

- عند استدعاء الطريقة - تحتك معامل محرر العنوان - <sup>أجل</sup> فيقول ما ، للإبني البرنامج نسخة عن المتول وإنما يستخدم المتول نفسه ( بحيث يؤثر ) أي تعديل على النسخة الممررة ، مع المتول الأصلي .

- يجب الانتباه أن الاستدعاء للطريقة يجب أن يفهم أيضاً الكلمة المفتاحية ref .

تمرين [ تخطيط ] بمفكرة التمرير بالعنوان :

حاصر قيمة k التي تظهر نتيجة تنفيذ البرنامج التالي ؟

1. static void Main (string [] args)
2. {
3. int [] table = {12, -5, 7, 8, -6, 6, 4, 78, 2} ;
4. int k = 0 ;
5. short i ;
6. for (i = 0 ; i <= 8 ; i++)
7. Increment (ref k) ;
8. console.WriteLine ("Result: " + k) ;
9. }
10. static void Increment (ref int val)
11. {
12. val = val + 1 ;
13. }

Result: 9

مثال: تقوم الطريقة Increment بإضافة 1 إلى قيمة المتول إذا كان لياوي الصفر

```
1. static void Main (String[] args)
2. {
3.     int a = 0;
4.     a = Increment (a); Console.WriteLine ("a = " + a);
5.     a = Increment (a+4); Console.WriteLine ("a = " + a);
6. }
```

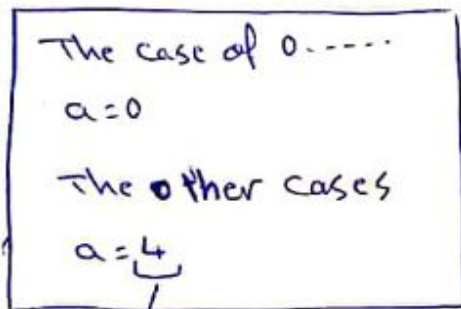
```
7. static int Increment (int x)
```

```
8. {
9.     if (x == 0)
10.    { Console.WriteLine ("The case of 0 : ----");
11.    }
12.    return x;
13. }
```

```
14. else
15.    { Console.WriteLine ("The other cases");
16.    }
```

```
17.    return x++;
18. }
```

زيادة ظرفية



لو كانت زيادة أصحبة لأصحبة  
القيمة 5

مدرى تعريف المتواليات :

• المتحول مرتب في ضمن الكتلة التي جرى تعريفه فيها .

• يمكن استخدام نفسه في المتحول في طريقتين مختلفتين لأنه يكون متحول محلي بالنسبة للفرضية المعرفه ضمنها .

تدريجه 1:

اكتب برنامج كاج مربعات الأعداد من 1 إلى 100 وذلك من خلال تعريفه لطريقة تدعى Square كاج المربع .

تدريب 2: أعط نتيجة تنفيذ البرنامج التالي وعلل النتيجة التي تظهر في كل مرة وبعد استدعاء الطريقة من الطرفين A و B.

1. using System

2. class Scoping

3. {

4. static int x = 1 ;

5. static void Main (String [] args)

6. {

7. Console.WriteLine ("Local x in method " + x) ;

8. Method A () ;

9. Console.WriteLine ("Local x in 1st call of A : " + x) ;

10. Method B () ;

11. Console.WriteLine ("Local x in 1st call of B : " + x) ;

12. Method A () ;

13. Console.WriteLine ("Local x in 2nd call of A : " + x) ;

14. Method B () ;

15. Console.WriteLine ("Local x in 2nd call of B : " + x) ;

16. }

17. static void Method A ()

18. {

19. int x = 25 ;

20. ++x ;

21. }

22. static void Method B ()

23. {

24. x \*= 10 ;

25. }

26. }

Local x in method 1
Local x in 1st call of A : 1
Local x in 1st call of B : 10
Local x in 2nd call of A : 10
Local x in 2nd call of B : 100

```

1. using System;
2. class Factorial
3. {
4.     static int x = 1;
5.     static void Main(string [] args)
6.     {
7.         string s; ushort a = 1000; long f;
8.         s = Console.ReadLine();
9.         a = UInt16.Parse(s);
10.        while (a != 0)
11.        {
12.            f = Fact(a);
13.            Console.WriteLine("Fact of " + a + " = " + f);
14.            s = Console.ReadLine();
15.            a = UInt16.Parse(s);
16.        }
17.    }
18.    static long Fact(ushort a)
19.    {
20.        long res = a;
21.        while (a > 1)
22.        {
23.            a--;
24.            res = res * a;
25.        }
26.        return res;
27.    }

```

$$u(n) = \begin{cases} n & \text{if } n=0 \text{ or } n=1 \\ u(n-1) + u(n-2) \end{cases}$$

الحب برنامج ايكير الطلب من المستخدم اذغال قيمه n فان الخط صحيح وضام سيدي الطريقة Fibonacci(n) بحيث تقوم هذه الطريقة بحساب قيمه u(n). (يتم البرنامج بالحل مع تكرار طلبات اذغال ارقام حتى يرضى المستخدم عدداً سالبا).

$$\begin{aligned} u(3) &= u(2) + u(1) \\ &= u(1) + u(0) + 1 \\ &= 1 + 0 + 1 = 2 \end{aligned}$$

1. class scoping

2. {

3. static int x=1 و

4. static void Main (String[] args)

5. {

6. string s;

7. int n;

8. long f;

9. s = console.ReadLine();

10. n = Int32.Parse(s);

11. while (n > 0)

12. {

13. f = Fibonacci(n);

14. console.WriteLine("u(" + n + ") = " + f);

15. s = console.ReadLine();

16. n = Int32.Parse(s);

17. }

18. }

19. static long Fibonacci (int number)

20. { if (number == 0 || number == 1)

21. return number;

22. else

23. return Fibonacci (number-1) + Fibonacci (number-2);

}



التمرين الأول: خوارزمية اقليدس.

اكتب برنامج لحساب القاسم المشترك الاكبر لعددين معينين لياوسان الصفر (خوارزمية اقليدس).

الطريقة 1: استخدم خوارزمية اسناد (قيمة باقى القسمة الاكبر على الصفر) اى (العدد الاكبر).

```
Static void Main(String[] args)
{
    Console.WriteLine("First Number: ");
    int p = Int32.Parse(Console.ReadLine());
    Console.WriteLine("Second Number: ");
    int q = Int32.Parse(Console.ReadLine());
    if (p * q != 0)
        Console.WriteLine("mgcd of " + p + " and " + q + " = " + mgcd(p, q));
    else
        Console.WriteLine("one of the numbers is null!");
}

Static int mgcd(int a, int b)
{
    int r;
    while ( (a != 0) && (b != 0) )
    {
        if (a > b)
            a = a % b;
        else
            b = b % a;
    }
    r = (a == 0) ? a : b;
    return r;
}
```

Milestone 4a  
م. خان حسين

تدريب: عدل على البرنامج السابق بحيث تستخدم طريقتي أوفزارزمية اسناد (قمتي حاصل  
طرح الأصغر من الأكبر) إلى العدد الأكبر.

التمرين الثاني: اكتب برنامجاً ليظهر أول  $n$  عدد أولي من مجموعة الأعداد الصحيحة الموجبة.

```
static void Main(string[] args)
{
    int divis, nbr, n, count = 0;
    bool isPrem;
    Console.WriteLine("How much numbers to Display?");
    n = Int32.Parse(Console.ReadLine());
    Console.WriteLine(2);
    nbr = 3;
    while (count < n-1)
    {
        divis = 2;
        isPrem = true;
        do
        {
            if (nbr % divis == 0)
                isPrem = false;
            else
                divis = divis + 1;
        }
        while ((divis <= nbr / 2) && (isPrem == true));
        if (isPrem)
        {
            count++;
            Console.WriteLine(nbr);
        }
        nbr++;
    }
}
```

التمرين الثالث : اكتب برنامجاً للتحقق من أن سلسلة محارف تمتلك صفة التناظر ، أي أنها تبقى نفسها سواء قرأناها من اليمين إلى اليسار أو من اليسار إلى اليمين .

مثال: acca, abcddeba

```
class program
```

```
{
```

```
    static string inverse(string s)
```

```
    {
```

```
        string r = " " ;
```

```
        int L = s.Length ;
```

```
        for (int i = 0 ; i <= L-1 ; i++)
```

```
            r = r + s[L-1-i] ;
```

```
        return r ;
```

```
    }
```

```
public static void Main (string[] args)
```

```
{
```

```
    string s ;
```

```
    Console.WriteLine ("Enter string : ") ;
```

```
    s = Console.ReadLine () ;
```

```
    string invs = inverse (s) ;
```

```
    Console.WriteLine ("Your string : " + s) ;
```

```
    Console.WriteLine ("Invers string : " + invs) ;
```

```
    if (s == invs)
```

```
        Console.WriteLine ("Palindrone ! ") ;
```

```
    else
```

```
        Console.WriteLine ("Not palindrone ! ") ;
```

العدد Armstrong هو عدد مساوياً كإجمالي جمع مكعبات الأرقام التي يتكونه

$$153 = 1^3 + 5^3 + 3^3$$
$$= 1 + 125 + 27 = 153$$

كل الأعداد Armstrong من مرتبة المئات .

الكتب برنامجاً لتدويرها .

```
static void Main(Strings[] args)
```

```
{  
    int i, j, k, n, sumCube;  
    Console.WriteLine("Number of Armstrong:");  
    for (i = 1; i <= 9; i++)  
        for (j = 0; j <= 9; j++)  
            for (k = 0; k <= 9; k++)  
            {  
                n = 100 * i + 10 * j + k;  
                sumCube = i * i * i + j * j * j + k * k * k;  
                if (sumCube == n)  
                    Console.WriteLine(n);  
            }  
}
```

اكتب برنامجاً يقوم بعملية بحث خطية تالية في عنصر  $x$  ضمن جدول  $T$  مؤلف من  $n$  عنصر  
ويُعطي ترتيبه في حال وجوده .

```
public static void Main (String [] args)
```

```
{
    int x ; string SX ;
    int [] T = {12, 10, 25, 16, 40, 45, 51, 60, 75, 90, 140, 120};
    int i, ord, N ;
    N = T.Length ;
    // Display Table
    for (i = 0 ; i <= N-1 ; i++)
        Console.WriteLine ("[" + i + "] | " + T[i]) ;
    // Read x
    Console.WriteLine ("choose number of the table:");
    SX = Console.ReadLine () ; x = Int32.Parse (SX) ;
    // Find x
    i = 0 ;
    while (i < N) && x != T[i]
        i = i + 1 ;
    if (i < N)
    {
        ord = i ;
        Console.WriteLine ("the element exists, order: " + ord) ;
    }
    else
        Console.WriteLine ("element not found") ;
}
```

الترتيب السارس: ليكن لدينا جدول T مرتب ترتيباً تصاعدياً ويحتوي على N عنصر،  
ولكن x عنصر من هذا الجدول، ما هي قيمة تنفيذ البرنامج التالي؟

```
int x; string sx;
```

```
int[] T = {12, 18, 25, 30, 40, 45, 51, 60, 75, 90, 100, 120};
```

```
int bottom, middle, top, ordre, N;
```

```
N = T.length;
```

```
for (int i = 0; i <= N-1, i++)
```

```
    Console.WriteLine("[ " + i + " ]\t" + T[i]);
```

```
Console.WriteLine("choose number of the table :");
```

```
sx = Console.ReadLine(); x = Int32.Parse(sx);
```

```
ordre = -1; bottom = 1; top = N-1;
```

```
do  
{
```

```
    middle = (bottom + top) / 2;
```

```
    if (x == T[middle],
```

```
        ordre = middle;
```

```
    else
```

```
        if (x > T[middle]
```

```
            bottom = middle + 1;
```

```
        else
```

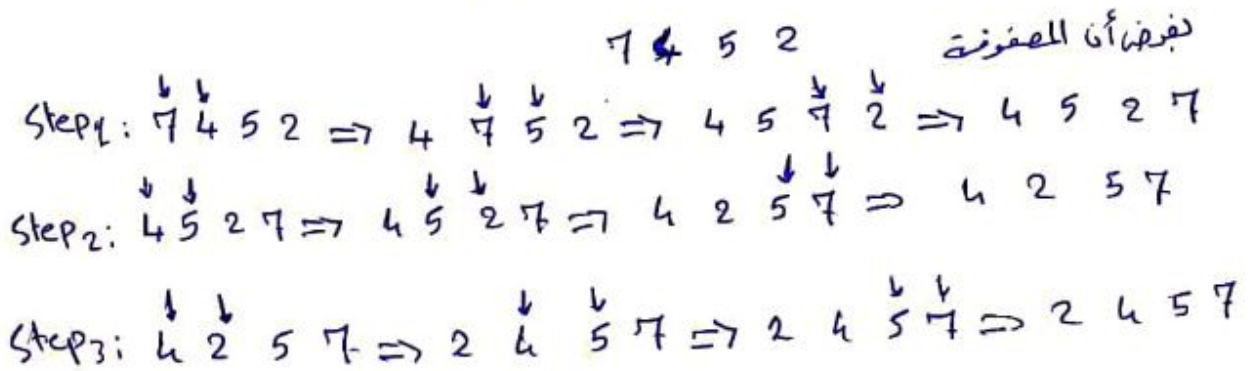
```
            top = middle - 1;
```

```
    }
```

```
    while (x != T[middle]);
```

```
Console.WriteLine("In ordre (index) of " + x + " = " + ordre);
```

الكتب برنامجاً قِراً جدولاً من  $n$  رقم ويرتبها بالترتيب التصاعدي أو التنازلي حسب خوارزمية <sup>التي</sup> Bubble Sort [الفرز الفقاعي].



```
static void Main(string[] args)
{
    int[] t = {7, 4, 5, 2};
    for (int i = 1; i <= t.Length; i++)
        for (int j = 1; j < t.Length; j++)
        {
            if (t[j-1] > t[j])
            {
                int temp = t[j-1];
                t[j-1] = t[j];
                t[j] = temp;
            }
        }
    for (int i = 0; i < t.Length; i++)
        Console.WriteLine(t[i]);
}
```

التمرين الثامن، اكتب برنامجاً لقراءة جدولاً من  $n$  رقم ورتبها بالترتيب التصاعدي أو التنازلي  
صه ضرورية أكثر  
Sort by Insertion

```
int [] t = {4, 2, 7, 5};  
for (int i = 1; i < t.Length; i++)  
    for (int j = i; j >= 1; j--)  
        if (t[j] < t[j-1])  
            {  
                int temp = t[j-1];  
                t[j-1] = t[j];  
                t[j] = temp;  
            }  
    }  
for (int i = 0; i < t.Length; i++)  
    Console.WriteLine(t[i]);
```

