بسم الله الرحمن الرحيم

النسخة الثانية ٢٠١٢/١١

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

الفهرس

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

المقدمة:

الحمد لله رب العالمين والصلاة والسلام على سيد الخلق أجمعين وبعد

قمت بطباعة هذا الكتاب في لغة البرمجة c++ نظراً للحاجة الملحة لكتاب يساعد زملائي الطلاب والمبتدئين في فهم البرمجة باستخدام هذه اللغة

وعلى الرغم من كثرة المراجع في هذا المجال إلا أنني لم أجد المرجع المناسب الذي يتوافق مع مقرر جامعة ذمار وذلك لأن المؤلفين لتلك المراجع يدرسونها في جامعات أخرى ودول أخرى وفي مراحل دراسية مختلفة

لذلك قمت بجمع وشرح البرامج التي درستها في مستوى أول وقد حاولت بذل ما أستطيع من الجهد في سبيل تحقيق الفائدة العامة والخاصة لي ولزملائي الطلاب
بالإضافة الى بعض أسئلة امتحانات مقرر المادة في العملي والنظري
ولا أمانع من التعديل على هذا الكتاب أو الإضافة اليه من قبل أي شخص يملك الخبرة الكافية في لغة c++ وذلك لتطوير النسخة القادمة من هذا الكتاب وعلى كل من يجد في رأيه تعديل أو إضافة الى هذا الكتاب التواصل معي على الرقم التالي : ٧١٥٤٣٢٣٣٧

أو البريد الإلكتروني: mazen_rawna2007@yahoo.com

هذا الكتاب ليس مجاني

اسم الكتاب: الطريق السهل لاحتراف البرمجة بلغة **c++**

الكاتب: مازن عباس عبدالله الرونة

المهنة: طالب في جامعة ذمار- كلية الحاسبات ونظم المعلومات - مستوى ثاني - قسم العلوم

سعر الكتاب : **دعوة صالحة في ظهر الغيب**

الإهداء:-

أهدي هذا العمل المتواضع الى:

- والدي ووالدتي أطال الله بقائهما
- أساتذة مادة البرمجة

د. خالد الحسيني (مدرس النظري)

أ. خالد الطاهري (مدرس العملي- الفصل الثاني)

أ. ياسمين المخلافي(مدرس العملي-الفصل الأول)

- كادر التدريس في كلية الحاسبات
- الى كل من علمني حرفاً
- زملائي الطلاب والطالبات
- كل من يحاول تعلم البرمجة

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

"لمحة تاريخية":-

لا يخفى على المتتبع لتطور لغات البرمجة أنها – ومنذ بناء أول حاسوب الكتروني – قد مرت بمراحل تطور هائلة.
فمنذ الحرب العالمية الثانية تم استخدام الحاسوب لإجراء العمليات الحسابية المعقدة فقد استخدمت لغة الآلة بإيعازاتها الأولية لكتابة أول برنامج يقوم بهذه الحسابات وكانت الإيعازات تكتب بلغة الأصفار والواحدات (zeros & ones) وبعدها بفترة قصيرة تم تطوير لغة الآلة الى لغة كانت إيعازاتها أقرب الى اللغة الإنجليزية مما ساعد على سهولة كتابة البرامج المعقدة أطلق على اللغة الجديدة(لغة التجميع)
وبعدها ظهر جيل جديد من اللغات مثل لغة (BASIC) ولغة (BASCAL) أطلق عليها لغات المستوى العالي
وفي سبعينيات القرن العشرين ظهرت لغة C لتشكل مع لغة (BASCAL) أسلوباً جديداً في كتابة البرامج أطلق عليها بالبرمجة المهيكلة
وفي عام ١٩٨٠ تم تصميم لغة C++ المنبثقة من لغة C وأصبحت لغة C++ من أهم اللغات واسعة الانتشار في ذلك الوقت وتعتبر لغة C++ الجسر الرابط بين لغات المستوى الواطئ.

عشر نصائح للنجاح في مادة البرمجة بلغة c++ :

١- اجعل من البرمجة مادة ممتعة تلجأ اليها في وقت الفراغ أو كلما أصبت بالضجر ولابد من فهم موضوعات البرمجة أولاً بأول.
٢- لا تقل لا أستطيع بل قل سأحاول مهما كان البرنامج الذي طلب منك صعباً.
٣- عند البدء في البرنامج ركز على المسألة المطلوب حلها وافهمها جيداً وقم بتجزئة المشكلة الكبيرة الى عدة مشاكل صغيرة ثم قم بحل كل مشكلة صغيرة على انفراد.
٤- ركز على طريقة حل المسألة واذا تعددت الأدوات لا تهتم أي الأدوات ستستخدم اذا كانت تؤدي الغرض المطلوب ولا تهتم بطريقة كتابة البرنامج بل ركز على حل المسألة ببساطة
٥- اذا وجدت صعوبة في حل مسألة ما حاول مرات أخرى وبطرق مختلفة وإذا أخبرك المترجم بخطأ ما حاول معرفة الخطأ واجعل من المحاولات متعة وتشويق لا اكراه.
٦- اذا عجزت عن حل مسألة عجزاً تاماً فلا مانع من إستشارة زملائك أو من هم أكثر خبرة منك ولا تأخذ شيء لم تفهمه بل ناقش الشيء حتى تفهمه ولا تتخذ مساعدة أحدهم ملاذاً لك كلما طلب منك كتابة برنامج بل يجوز المساعدة في حالات الضرورة فقط وحاول الاعتماد على نفسك في كل شيء.
٧- اذا وجدت طريقة أخرى للحل لا تترك طريقتك لأن ذلك يتسبب في فقدان الثقة بالنفس فلكل مبرمج طريقته في البرمجة.
٨- لا تتردد في كتابة وتجربة أي برنامج يخطر في بالك مهما كان صغيراً أو كبيراً فلا يوجد أفكار ساذجة أو مستحيلة فهذا أسهل طريق للإبداع.
٩- ثق بقدراتك ولا تسمح لأي شيء أن يهز ثقتك بنفسك.
١٠-نتائج الامتحانات لا تعبر بالضرورة عن مستويات الطلاب - خاصة في البرمجة- بل مقدار العمل والثقة بالنفس والاعتماد على النفس وأخيراً القدرة على حل المسائل هي التي تحدد مستويات الطلاب.

المتغيرات وأنواع البيانات في لغة c++

عند تعلم أي لغة جديدة- طبيعية كانت أم لغة حاسوب – فإن ذلك يتطلب معرفة أساسيات وقواعد تلك اللغة مثل الحروف الأبجدية وقواعد القراءة والكتابة وكيفية دمج الحروف مع بعها لتكوين الكلمة وكيفية صياغة الجمل

- أساسيات لغة البرمجة c++

١- الحروف letters: تشمل حروف الأبجدية الإنجليزية الكبيرة والصغيرة {A,B,C,,,,Z,a,b,c,,,,z}

٢- الأرقام numbers: تشمل الأرقام العشرية {0,1,2.....,9}

٣- الرموز الخاصة special characters: تشمل الرموز الموضحة في الجدول التالي

( )	| |	%	>	,	+
{ }	"	&	>=	:	-
[ ]	'	$	<	;	*
>>	?	#	<=	.	/
<<	\	~	<>	^	=
!=		!	..	↑	→

٤- الكلمات المحجوزة reserve words: وهي كلمات معرفة مسبقاً لدى مترجم لغة c++ ولا يجوز للمبرمج استخدام هذه الكلمات في غرض غير المخصص لها

asm	continue	far	near	return	typedef	extern
auto	default	float	new	short	union	long
break	delete	for	operator	signed	unsigned	do
case	register	friend	overload	sizeof	virtual	this
catch	double	goto	pascal	static	void	
cdecl	else	if	private	struct	volatile	
char	entry	inline	protected	switch	while	
class	enum	int	public	template	const	

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

- المتغيرات variables: المتغيرات هي أسماء لمواقع في الذاكرة تستخدم للدلالة على قيمة معينة تستخدم داخل البرنامج وكل متغير يجب أن يعلن عنه قبل إستخدامه في البرنامج

-الإعلان والتصريح عن المتغيرات :-

يتم الإعلان عن المتغيرات أولاً بتحديد نوع المتغير هل هو من النوع الحقيقي real numbers أو الصحيح integer numbers أو الطبيعي natural numbers أو الحرفي characters وغيرها من الأنواع ولكل نوع حجم محدد يمكن معرفته من خلال العامل**sizeof(type)**

ثم بتسمية المتغير بإسم يخضع لشروط التسمية التالية :-

١- أن يبدأ بأحد الحروف الأبجدية الكبيرة أو الصغيرة أو الرمز _ underscore فلا يجوز أن يبدأ الاسم برقم

٢- أن لا يحتوي إسم المتغير على رمز من الرموز مثل (؟ / * - > < + : " ~ ! @ # $ % &) وغيرها من الرموز

٣- أن لا يحتوي اسم المتغير على الفراغ

٤- أن لا يزيد طول اسم المتغير عن ٣٢ رمز

٥- أن لا يكون إسم المتغير من الكلمات المحجوزة

الاستخدام	المثال	التعبير
للإعلان عن متغيرات من النوع الصحيح	int x;	int
للإعلان عن متغيرات من النوع الكسري	float x;	float
	double x;	double
للإعلان عن متغيرات من النوع الصحيح أكبر من ٣٢٧٠٠	long int x;	long int
للإعلان عن متغيرات من النوع الكسري أكبر من ٣٢٧٠٠	long float x;	long float
للإعلان عن المتغيرات الرمزية	char x;	char

أمثلة عن التصريح عن المتغيرات

أ- الإعلان عن متغير من النوع الصحيح;int x ب-من النوع الكسري;float x

ج- من النوع الصحيح أكثر من ٣٢٧٠٠ ;long int x

د- من النوع الكسري أكثر من ٣٢٧٠٠ ;long float x

أمثلة إضافية الإدخال المباشر

1- int x=5; 2- float x=7.5 3- long int x=50000
4- long float x=50000.5 5- double x=1000

أما اذا كان المتغير ثابت ولا نريد تغيير قيمته أبداً فيتم الاعلان عنه باستخدام الكلمة const ثم تحديد نوع المتغير وعند تغيير قيمته بأي شكل لا يتم تنفيذ البرنامج ويتم الاعلان عن الثابت بايπ كما يلي const float pi=3.14

- الشكل العام للبرنامج المكتوب بلغة c++

```
#include<header files.h>
main()
{program body}
```

حيث include:هي جمله لتضمين المكتبات (ملفات الترويسة header files) والمكتبات تستخدم لتعريف الدوال المستخدمة في البرنامج و main: هي الدالة الرئيسة التي يكتب البرنامج بداخلها والتي يتم تحميلها الى الذاكرة الرئيسية ram عند تشغيل البرنامج أو العمل عليه أما program body: هو جسم البرنامج أي الدوال والتعليمات التي يكتبها المبرمج ليقوم البرنامج بالعمل المطلوب والجدول التالي لتوضيح أهم المكتبات واستخدامها

اسم المكتبة	استخدامها
iostream	لتعريف دوال الإدخال والإخراج input & output functions
conio	لتعريف دوال نظام dos
math	لتعريف دوال الرياضيات
string	لتعريف دوال السلاسل الحرفية
stdio	لتعريف دوال الإدخال والإخرج الخاصة بلغة c
graphics	لتعريف دوال الرسومات
time	لتعريف دوال الوقت

وسميت المكتبة iostream بهذا الإسم حيث input=i وoutput=o وstream تعني نهر أي نهر تعليمات الإدخال والإخراج ومن دوال الإدخال والاخراج التابعة لمكتبة iostreamهي دالة الإدخال cin ودالة الإخرج coutومن الدوال التابعة لمكتبة conio دالة تثبيت شاشة المخرجات ()getch وكذلك دالة مسح الشاشة ()clrscr ومن الدوال التابعة لمكتبة math دالة القوة pow(x,y) وكذلك دالة الجذر التربيعي sqrt(x) ومن الدوال التابعة لمكتبة stdio دالة الإدخال scanf وكذلك دالة الاخراج printf

- دوال الإدخال والإخراج في لغة c++

تستخدم دوال الإدخال لإدخال قيمة متغير معين ثم إجراء العمليات الحسابية والمنطقية عليه ودوال الإدخال هي cin للإدخال غير المرتب وتقع هذه الدالة ضمن المكتبة iostream ويكون الشكل العام لهذه الدالة كما يلي
cin>>variable حيث variable هو إسم متغير معين وكذلك الدالة scanf وتقع ضمن المكتبة stdio والدالة التي تهمنا في الإدخال هي الدالة cin أما في الإخراج(الطباعة) سنستخدم الدالة cout وشكلها العام كما يلي
cout<<variable ولطباعة الجمل التعريفية نستخدم الدالة cout<< ثم نكتب الجمل المراد طباعتها على شاشة المخرجات بين علامتي تنصيص مزدوجة "mazen abas"

ملاحظة: لابد أن تنتهي كل تعليمة (instruction) في البرنامج المكتوب بلغة c++ بفارزة منقوطة (;)

*أول برنامج ويقوم بطباعة أي جملة "welcome to c++ world"

```
#include<iostream.h>
main()
{cout<<"welcome to c++ world";}
```

أولاً قمنا بتضمين المكتبة iostream لتعريف الدالة cout ثم فتحنا الدالة الرئيسية main وكتبا البرنامج وهو دالة cout ثم الجملة المطلوب طباعتها بين علامتي تنصيص مزدوجة ثم أنتهى البرنامج بفارزة منقوطة وأغلقنا الدالة الرئيسية main

وعند تنفيذ البرنامج على المترجم تكون المخرجات كما في الشكل التالي

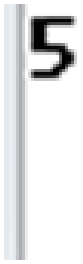
```
welcome to c++ world
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يقوم بقراءة رقم صحيح وطباعته والإدخال مباشر(عند التعريف)

```
#include<iostream.h>
main()
{int x=5;
cout<<x;}
```

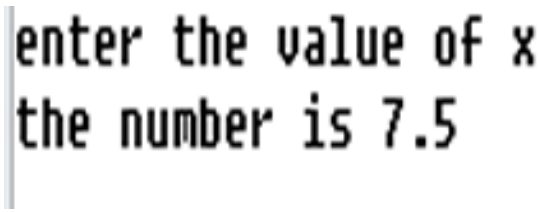
وعند تنفيذ الكود على المترجم تكون المخرجات كما يلي:



*برنامج لإدخال رقم حقيقي عن طريق المستخدم وطباعته

```
#include<iostream.h>
main()
{float x;
cout<<"enter the value of x";
cin>>x;
cout<<"the number is "<<x;}
```

أولاً قمنا بتعريف عدد x من النوع الحقيقي ثم قمنا بطباعة جملة تعريفية تأمر المستخدم بإدخال قيمة x بإستخدام الدالة cout ثم استخدمنا جملة الإدخال cin لإدخال قيمة المتغير x ثم استخدمنا جملة الطباعة cout لطباعة قيمة x المدخلة وعند إدخال القيمة ٧.٥ تكون المخرجات كما يلي:



مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يقوم بقراءة حرف من الأبجدية الإنجليزية وطباعته

```
#include<iostream.h>
main()
{char x='A';
cout<<x;}
```

عند تنفيذ البرنامج على المترجم تكون المخرجات كما يلي

A

وبنفس الطريقة في حالة الإدخال عن طريق المستخدم

ـ المعاملات الحسابية والمنطقية ومعاملات الزيادة والنقصان والإسناد:

المعاملات الحسابية في لغة c++ هي نفس معاملات الرياضيات حيث يستخدم المعامل "+" للجمع مثل x=a+b والمعامل "-" يستخدم للطرح مثل x=a-b والمعامل "*" يستخدم للضرب x=a*b والمعامل "/" يستخدم للقسمة مثل x=a/b وفي جميع الحالات السابقة استخدمنا معامل الإسناد"=" لإسناد القيم للمتغير x أما المعاملات المنطقية هي أكبر من مثل a>b حيث a أكبر من b وكذلك المعامل أصغر من مثل a<b حيث a أصغر من b

أما معاملات الزيادة مثل a++ و ++a و a+=1 و a=a+1 وفي جميع الحالات السابقة الزيادة بمقدار ١ أما a+=2 و a=a+2 وتكون الزيادة بمقدار ٢ وهكذا

جدول العمليات الحسابية والأدوات المستخدمة فيها

العملية	الأداة
للجمع	+
للطرح	-
للقسمة	/
باقي القسمة	%
للضرب	*
للزيادة بمقدار واحد	++
للنقصان بمقدار واحد	--

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

أولوية تنفيذ العمليات الحسابية

١- الأقواس
٢- الأسس
٣- الضرب والقسمة وباقي القسمة أيهما تأتي أولاً من اليسار
٤- الجمع والطرح أيهما يأتي أولاً من اليسار

المعاملات العلاقية

العملية	الرمز الرياضي	الرمز في الحاسوب
Xأكبر من Y	X>Y	X>Y
X أصغر من Y	X<Y	X<y
X أكبر من أو يساوي Y	X≥Y	X>=Y
X أصغر من أو يساوي Y	X≤Y	X<=Y
X يساوي Y	X=Y	X=Y
X لا يساوي Y	X≠Y	X!=Y

أما أولوية العمليات العلاقية فهي بنفس ترتيب الجدول من أعلى الى أسفل

المعاملات المنطقية

العملية	الأداة(الرمز)
AND	&&
OR	||
NOT	!

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

المعاملات الخاصة بالبت:

الحدث	العامل (الرمز)
AND	&
OR	|
XOR	^
ONE'S COMPLEMENT	~
SHIFT RIGHT	>>
SHIFT LEFT	<<

الثوابت الرمزية المستخدمة في عمليات الاخراج

العمل والوصف	الرمز
ارجاع القيمة المطبوعة مسافة واحدة الى الخلف	\b
للنزول الى سطر جديد	\n
يزيح القيمة(٨ مسافات أفقية) بعد آخر عملية طباعة	\t
طباعة علامة الاقتباس المزدوجة	\"
طباعة علامة الاقتباس المفردة	\'
طباعة خط مائل	\\
صوت تنبيهي	\a
لإظهار علامة الاستفهام	\?

*برنامج يقوم بإدخال عددين a,b عن طريق المستخدم ثم يجمعهما ويسند ناتج الجمع للمتغير c ويطبع ناتج الجمع

```
#include<iostream.h>
main()
{int a,b,c;
cout<<"enter a : ";
cin>>a;
cout<<"enter b : ";
cin>>b;
c=a+b;
cout<<"the sum is "<<a<<" + "<<b<<"= "<<c;}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يقوم بإدخال عددين x&y ثم يجمعهما ويضربهما ويطبع وينقص من قيمة x بمقدار ١ ويزيد قيمة y بمقدار ١ ويطبع القيمتين الجديدة

```
#include<iostream.h>

main()
{int x,y;
cout<<"enter x : ";
cin>>x;
cout<<"enter y : ";
cin>>y;
cout<<"x+y="<<x<<"+"<<y<<" = "<<x+y;
cout<<"\nx*y="<<x<<"*"<<y<<" = "<<x*y;
x--;
y++;
cout<<"\nthe new x= "<<x;
cout<<"\nthe new y= "<<y;}
```

-الجمل الشرطية(if -if else-if else if- switch)

١- جملة الشرط if:إذا تحقق الشرط يتم تنفيذ ما بعد الشرط وإذا لم يتحقق ينتقل المعالج الى التعليمة التالي والشكل العام هو if(condition) statement;

حيث if : جملة الشرط و condition :الشرط المطلوب تحققه لتنفيذ ما بعده و statement: التعليمة التي يتم تنفيذها إذا تحقق الشرط

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

مثال: برنامج لإدخال رقم صحيح وإذا كان الرقم أكبر من ١٠ يطبع large وإن لم يكن كذلك يتوقف البرنامج

```
#include<iostream.h>
main()
{int y;
cout<<"enter the value of y : ";
cin>>y;
if(y>10)
cout<<"large";}
```

يمكن أن يتكرر استخدام if بقدر الحاجة في برنامج واحد كما في المثال وهو برنامج يقوم بإدخال رقم ويختبرأذا كان أصغر من١٠يطبع "small" وإذا كان الرقم يساوي ١٠ يطبع "equal" وإذا كان الرقم أكبر من ١٠ يطبع "large" وعند تعدد جمل ifيتم اختبار جميع الشروط ولا يتوقف التنفيذ عند تحقق أحدها

```
#include<iostream.h>
main()
{int y;
cout<<"enter the value of y : ";
cin>>y;
if(y<10)  cout<<"small";
if(y==10) cout<<"equal";
if(y>10) cout<<"large"; }
```

إذا تحقق شرط if يتم تنفيذ التعليمة التي تليها الى الفارزة المنقوطة ; (تعليمة واحدة فقط) وإذا أردنا تنفيذ عدة تعليمات معاً فإننا نجعلها داخل حاصرتين {instructions}

وكذلك يمكن أن تتداخل جمل if في نفس البرنامج وذلك في حالة تعدد الشروط والمثال التالي وهو

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يقوم بإدخال عدد صحيح ويختبر شرطين الأول إذا كان عدد موجب وإذا تحقق الشرط الأول يختبر الشرط الثاني وهو هل هو عدد زوجي إذا تحقق الشرطان يطبع جملة "true"

```
#include<iostream.h>
main()
{int x;
cout<<"enter the number : ";
cin>>x;
if(x>=0)
{if(x%2==0)
cout<<"true";}}
```

ويمكن الاستغناء عن if المتداخلة بإستخدام المعامل && ويكون الشرط كما يلي

```
if(x>=0 && x%2==0)
```

٢- جملة الشرط (if else):تقوم هذه الجملة الشرطية باختبار شرط معين إذا تحقق الشرط يتم تنفيذ التعليمة التي بعد if وإذا لم يتحقق الشرط يتم تنفيذ التعليمة التي بعد else

الشكل العام لجملة (if else)

```
if(condition)
statement1
else  statement2
```

حيث condition : شرط معين و statement1: التعليمة التي يتم تنفيذها إذا تحقق الشرط وstatement2: التعليمة التي يتم تنفيذها إذا لم يتحقق الشرط ويكون عدد مرات استخدام if في حالة استخدام else بعدد الشروط مطروح منه واحد والبرنامج التالي مثال لجملة (if else)

*برنامج يقوم بإخال عدد صحيح ويختبر إذا كان العدد زوجي يطبع "even"وإذا كان فردي يطبع"odd"

```
#include<iostream.h>
main()
{int x;
cout<<"enter the number : ";
cin>>x;
if(x%2==0)
cout<<"even";
else cout<<"odd";}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يقوم بإدخال عدد ويختبر العدد إذا كان موجب يطبع"positive" وإذا كان العدد سالب يطبع"negative"

```
#include<iostream.h>
main()
{int x;
cout<<"enter the number : ";
cin>>x;
if(x>=0)
cout<<"positive";
else cout<<"negative";}
```

٣- جملة الشرط (if else if) تشبه (if else) تقريباً في عملها والشكل العام لجملة(if else if) كما يلي:

```
if(condition1)   statement1;
 else if(condition2)   statement2;
 else  statement3;
```

حيثcondition1 هو الشرط الأول وstatement1 هي التعليمة التي يتم تنفيذها إذا تحقق الشرط الأول وcondition2 هو الشرط الثاني وstatement2 هي التعليمة التي يتم تنفيذها إذا تحقق الشرط الثاني أما statement3 فهي التعليمة التي تنفذ إذا لم يتحقق أحد الشرطين السابقين والبرنامج التالي مثال لجملة(if else if)

*برنامج يقوم بإدخال عددين x & y ثم يقارن بين العددين هل هما متساويان أم أن أحدهما أكبر من الآخر

```
#include<iostream.h>
main()
{int x,y;
cout<<"enter the value of x : ";
cin>>x;
cout<<"enter the value of y : ";
cin>>y;
if(x>y)
cout<<"x is larger then y because "<<x<<">"<<y;
else if(x<y)
cout<<"x is smaller then y because "<<x<<"<"<<y;
else cout<<"x is equal y because "<<x<<"="<<y;}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

٤- جملة الشرط(الإختبار)(switch) :إن عمل جملة switch مشابه لعمل جملة (if else) مع قليل من السهولة التي ستظهر في بعض الأمثلة أما الشكل العام لجملة switch فهو كما يلي

```
switch(variable)
{case 1:statement1;break;
case2:statement2;break;
default:statement3;break;}
```

حيثvariable متغير من النوع الصحيح أو الحرفي وcase1 هي شرط أو قيمة معينة للمتغير وstatement1 هي التعليمة التي ستنفذ في حالة تحقق الشرط الأولcase1 و default فهي شرط معناه إذا لم يتحقق أي من الشروط السابقة أماbreak فهي تستخدم للخروج من جملة switch في حالة تحقق أي حالة من الحالات السابقة والأمثلة كما يلي

*برنامج يقوم بإدخال عدد صحيح من ١ الى ٩ عن طريق المستخدم ويطبع الرقم بالحروف الإنجليزية وإذا كان خارج هذا المدى يطبع"undefined"

```
#include<iostream.h>
main()
{int x;
cout<<"enter the value of x : ";
cin>>x;
switch(x)
{case 1:cout<<"one";break;
case 2:cout<<"two";break;
case 3:cout<<"three";break;
case 4:cout<<"four";break;
case 5:cout<<"five";break;
case 6:cout<<"six";break;
case 7:cout<<"seven";break;
case 8:cout<<"eight";break;
case 9:cout<<"nine";break;
default:cout<<"undefined";break;}}
```

هناك أداة شرط أخرى وهي كما يلي

```
condition ?
statement1 : statement2;
```

إذا تحقق الشرط يتم تنفيذ statement1 وإذا لم يتحقق الشرط يتم تنفيذ statement2 والبرنامج التالي مثال على هذه الأداة

*برنامج يقوم بإدخال عددين x & y ثم يقارن بينهما أيهما أكبر

```
#include<iostream.h>
main()
{int x,y;
cin>>x>>y;
x>y? cout<<"x > y" : cout<<" y > x";}
```

***الدوارات loops :-** الدوارات هي جمل تستخدم لتكرار إجراء محدد يرغب المبرمج في تكراره وهي ثلاث جمل هي (for – while – do while)

١- الدوارة for :- ولها حد أدنى وحد أعلى ومقدار زيادة أو نقصان محدد وشكلها العام كما يلي:

```
for(i=a;i<b;i++)
statement;
```

حيث a هو الحد الأدنى وb هو الحد الأعلى وstatement هي التعليمة المرغوب في تكرارها و++i مقدار الزيادة بمقدار ١ وقد تكون الزيادة بمقدار ١ بعدة أشكال منها ++i أو i+=1 أو i++ وكذلك النقصان بمقدار ١ -- i أوi-=1 أو i -- وقد تكون الزيادة بمقدار ٢ أو ٣ أو بمقدار ما تريد والبرنامج التالي مثال للدوارة for وهو برنامج يقوم بطباعة الأعداد الزوجية من ١ الى ١٠٠

```
#include<iostream.h>
main()
{for(int i=1;i<=100;i++)
if(i%2==0)
cout<<i<<"  ";}
```

في المثال السابق الحد الأدنى هو١ حيث i=1 والحد الأعلى هو١٠٠ لأن i<=100 ومقدار الزيادة هو١ حيث ++i ويبدأ البرنامج من ١ ويختبر هل العدد زوجي أم فردي فلا يطبع الرقم ١ ثم يزداد بمقدار ١ويكون i=2 ويختبر العدد ٢ فهو زوجي فيطبعه وهكذا حتى يصل إلى ١٠٠ وتكون شاشة المخرجات كما يلي:

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

```
2  4  6  8  10  12  14  16  18  20  22  24  26  28  30  32  34  36  38  40  42
44  46  48  50  52  54  56  58  60  62  64  66  68  70  72  74  76  78  80  82
84  86  88  90  92  94  96  98  100
```

***برنامج يسمح بإدخال ١٠ أرقام ويختبر كل عدد هل هو زوجي أم فردي**

```
#include<iostream.h>
main()
{int x,i;
for(i=1;i<=10;i++)
{cout<<"enter the number : ";
cin>>x;
if(x%2==0)
cout<<"even\n";
else cout<<"odd\n";} }
```

وكما هو واضح في المثال السابق إذا أردنا تكرار أكثر من تعليمة لابد من وضع التعليمات المطلوب تكرارها بين حاصرتين{instructions} ويمكن أن تتداخل الدوارة for وعندئذ تسمى for المتداخلة كما في البرنامج التالي:

***برنامج يطبع الأرقام من ١ الى ١٠ خمس مرات كل مرة في سطر جديد بإستخدام for المتداخلة**

```
#include<iostream.h>
main()
{for(int i=0;i<=4;i++)
{for(int j=1;j<=10;j++)
{cout<<j<<"  ";}
cout<<"\n";} }
```

يظهر في البرنامج السابق أن الدوارة الثانية for(int j=1;j<=10;j++) تقوم بطباعة الأعداد من ١ الى ١٠ في كل مرة تنفذ الدوارة الأولى حيث أن الدوارة الثانية تابعة للأولى وعند تنفيذ الكود السابق على المترجم تكون المخرجات كما في الشكل التالي:

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

```
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
1  2  3  4  5  6  7  8  9  10
```

*برنامج يطبع الأرقام من ١٠٠ الى ١ تنازلياً

```
#include<iostream.h>
main()
{int i;
for(i=100;i>0;i--)
cout<<i<<"  ";}
```

عند تنفيذ الكود السابق تكون شاشة المخرجات كما يلي

```
100  99  98  97  96  95  94  93  92  91  90  89  88  87  86  85  84  83  82  81
 80  79  78  77  76  75  74  73  72  71  70  69  68  67  66  65  64  63  62  61
 60  59  58  57  56  55  54  53  52  51  50  49  48  47  46  45  44  43  42  41
 40  39  38  37  36  35  34  33  32  31  30  29  28  27  26  25  24  23  22  21
 20  19  18  17  16  15  14  13  12  11  10  9  8  7  6  5  4  3  2  1
```

إدخال for في دوارة لا نهائية

```
for(;;)
{statement;}
```

٢- الدوارة while لها حد أدنى يتم تحديده قبل الوصول الى الدوارة ولها حد أعلى ومقدر زيادة أو نقصان والشكل العام لها كما يلي

```
int j=a;
while(condition)
{statement;
j++;}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

حيث الحد الأدنى j=a و statement هي التعليمة أو التعليمات المطلوب تكرارها ومقدار الزيادة ١ ++j

*برنامج يطبع الأحرف الإنجليزية الكبيرة من A الى Z

```
#include<iostream.h>
main()
{char x='A';
while(x<='Z')
{cout<<x<<"  ";
x++;} }
```

عند تنفيذ الكود على المترجم تكون المخرجات كما يلي:

*برنامج للتحويل من النظام العشري الى النظام الثنائي باستخدام while

```
#include<iostream.h>
main()
{int x;
cin>>x;
while(x>0)
{cout<<x%2;
x=x/2;}}
```

إدخال while في دوارة لا نهائية يتم بالطريقة التالية

```
int a=1;
while(a>0)
{statement; a++;}
```

ملاحظة: تستخدم التعليمة ;break للخروج من أي دوارة كما في البرنامج التالي:

***برنامج الحاسبة يسمح بإجراء عدد لا نهائي من العمليات الحسابية باستخدام while اللانهائية**

```
#include<iostream.h>
#include<conio.h>
main()
{int i=1,x,y;
char a,b;
while(i>0)
{cout<<"enter the first number : ";
cin>>x;
cout<<"enter the operation : ";
cin>>a;
cout<<"enter the second number : ";
cin>>y;
 switch(a)
{case '+':cout<<x+y<<"\n";break;
 case '-':cout<<x-y<<"\n";break;
 case '*':cout<<x*y<<"\n";break;
 case '/':cout<<x/y<<"\n";break;
 default: cout<<"undefined\n";break;}
 cout<<"do you want to continue ? (Y/N)";
cin>>b;
if(b=='Y')
{clrscr();continue;}
else {cout<<"goodboy";break;} } }
```

٣- الدوارة do-while: تقوم هذه الدوارة بنفس العمل الذي تقوم به while وما يميزها أنه يتم تنفيذ العمل المطلوب مرة واحدة على الأقل قبل إختبار الشرط والشكل العام لها كما يلي:

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


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يقوم بإدخال عدد وإذا كان أصغر من ١٠٠ يطبع الأرقام من الرقم المدخل الى ١٠٠ وإذا كان أكبر من ١٠٠ يطبع العدد نفسه فقط

```
#include<iostream.h>
main()
{int x;
cin>>x;
do
{cout<<x<<"  ";
x++;}
while(x<=100);}
```

- **المصفوفات arrays :** نوع من أنواع البيانات التي تحتوي على أكثر من عنصر وتحمل نفس الإسم بترقيم يبدأ من الصفر بدل من a,b,c,d,e نعرف مصفوفة من ٥ عناصر x[5] وعناصرها هي x[0],x[1],x[2],x[3],x[4]

أ- المصفوفات الأحادية

*برنامج لقراءة مصفوفة أحادية وطباعتها ادخال مباشر

```
#include<iostream.h>
main()
{ int x[5]={1,2,3,4,5},i;
for(i=0;i<5;i++)
cout<<x[i]<<"\t";}
```

1	2	3	4	5

الشرح:

عند التعريف نعرف المصفوفة كأي متغير اخر يخضع لقواعد التسمية ثم نحدد عدد عناصر المصفوفة جوارها بين علامتي [ ] ثم أدخلنا العناصر ادخال مباشر واستخدمنا دوارة تتنفذ بعدد عناصر المصفوفة وتطبع عنصر في كل دورة ثم"\t" للازاحة الأفقية وعند التنفيذ تكون شاشة المخرجات كما في الشكل المجاور للكود

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يقوم بإدخال مصفوفتين أحاديتين إدخال مباشر ثم يجمعهما ويطبع مصفوفة ناتج الجمع

```
#include<iostream.h>
main()
{int x[5]={1,2,3,4,5} ,y[5]={1,2,3,4,5} , i, m[5];
for(i=0;i<5;i++)
{m[i]=x[i]+y[i];
cout<<m[i]<<"\t";} }
```

2	4	6	8	10

الشرح: قمنا بتعريف ثلاث مصفوفات أحادية x وyوmوعرفنا كذلك i الذي استخدمناه عداد داخل الدوارة ثم استخدمنا دوارة لإدخال عناصر المصفوفة الأولى ودوارة لإدخال عناصر المصفوفة الثانية ودوارة ثالثة خاصة بالمصفوفة m التي تجمع عناصر المصفوفتين الأولى والثانية وتطبع مصفوفة ناتج الجمع وبتغيير اشارة الجمع يمكن اجراء العمليات الحسابية البسيطة على المصفوفات للطرح نستخدم – وللضرب * وللقسمة /

وعند ادخال الارقام من ١-٥ لكل من المصفوفتين الاولى والثانية تكون شاشة المخرجات كما في الشكل المجاور للكود

*برنامج يقوم بقراءة مصفوفتين أحاديتين وينسخ الأولى الى الثانية

```
#include<iostream.h>
main()
{ int x[9]={1,2,3,4,5,6,7,8,9},y[9]={12,3,6,2,1,4,6,8,64},i;
for(i=0;i<9;i++)
{ y[i]=x[i];
cout<<y[i]<<"\t";} }
```

1	2	3	4	5	6	7	8	9

الشرح:

يقوم هذا البرنامج بإدخال مصفوفتين تتكون كل منهما من ٩ عناصر اسم الاولى x والثانية y ادخال مباشر وعناصر y تختلف عن عناصر x ثم يقوم بنسخ عناصر x الى y وعندما نطبع عناصر y تكون نفس عناصر x عند تنفيذ البرنامج تكون شاشة التنفيذ كما في الشكل أعلاه

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يقوم بقراءة مصفوفة أحادية ثم يطبع المصفوفة ويطبع مجموع عناصر المصفوفة

```
#include<iostream.h>
main()
{ int x[5]={1,2,3,4,5},i,y=0;
for(i=0;i<5;i++)
{ y=y+x[i];
cout<<x[i]<<"\t";}
cout<<"\n the sum="<<y;}
```

```
1    2    3    4    5
 the sum=15
```

الشرح:

قمنا بتعريف مصفوفة أحادية تتكون من ٥ عناصر وأدخلناها إدخال مباشر وعرفنا كذلك العداد i للدوارة وعرفنا y بقيمة تساوي (0) المحايد الجمعي وداخل الدوارة تتغير قيمة y في كل دورة حيث y=y+x[i] في الدورة الاولي y=0+1وفي الثاني y=1+2 وفي الثالثة y=3+3 وفي الرابعة y=6+4 وفي الخامسة y=10+5 ثم يقوم بطباعة عناصر المصفوفة x ويخرج من الدوارة ويطبع قيمة y ناتج جمع عناصر المصفوفة

ويكون ناتج تنفيذ البرنامج كما في الشكل المجاور للكود

وبتغيير قيمة y الى (1) المحايد الضربي وتغيير عملية الجمع الى ضرب يمكن ايجاد مضروب عناصر اي مصفوفة

```
#include<iostream.h>
main()
{ int x[5]={1,2,3,4,5},i,y=1;
for(i=0;i<5;i++)
{ y=y*x[i];
cout<<x[i]<<"\t";}
cout<<"\n the mul="<<y;}
```

```
1    2    3    4    5
the mul=120
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج لإدخال مصفوفة أحادية عن طريق المستخدم ويعكس ترتيب المصفوفة

```
#include<iostream.h>
main()
{int x[5],i;
cout<<"enter the components of array\n";
for(i=0;i<5;i++)
cin>>x[i];
cout<<"\n";
for(i=4;i>=0;i--)
cout<<x[i]<<"\t";}
```

5	4	3	2	1

الشرح:

عند التنفيذ على المترجم وإدخال الأرقام من ١- ٥ تصاعدياً الى قيم المصفوفة يكون شكل المخرجات كما في الشكل المجاور للكود

كل ما قمت بفعله هو أولاً بدأت قيمة العداد i من (٠-٤) ثم أثناء الإخراج عكست الدوارة حيث بدأت قيمة العداد i آخر موقع في المصفوفة الى أول موقع من (٤-٠)

*برنامج لقراءة مصفوفة أحادية مكونة من ١٠ عناصر من الأعداد الحقيقة أو الصحيحة وإيجاد أصغر قيمة في المصفوفة وطباعتها

```
#include<iostream.h>
main()
{int  x[10]={10,90,20,80,30,70,40,60,50,100},min,i;
min=x[0];
for(i=0;i<10;i++)
if(x[i]<min)
min=x[i];
cout<<min;}
```

10

الشرح: قمنا بإدخال مصفوفة من ١٠ عناصر إدخال مباشر وعرفنا متغيرmin ثم أسندنا له قيمة أول عنصر في المصفوفة ;min=x[0] وباستخدام الدوارة و if الشرط قمنا بمقارنة min بجميع عناصر المصفوفة عندما يكون أي عنصر أصغر من min ينقل قيمة العنصر للمتغير min وبعد انتهاء الدوارة طبعنا أصغر قيمة وهي الرقم (١٠)

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*وبعكس الشرط if(x[i]<min) الى if(x[i]>max) وبنفس الخطوات السابقة يمكن إيجاد أكبر عنصر في المصفوفة

*برنامج ترتيب مصفوفة أحادية تصاعدياً من الأصغر الى الأكبر

```
#include<iostream.h>
main()
{int x[10]={45,75,34,78,98,77,63,49,59,43},i,j,t;
for(i=0;i<10;i++)
for(j=0;j<10;j++)
if(x[i]<x[j])
{ t=x[j];  x[j]=x[i];  x[i]=t;}
for(i=0;i<10;i++)
cout<<x[i]<<"\t";}
```

34 43 45 49 59 63 75 77 78 98

*برنامج يقوم بإدخال مصفوفة عن طريق المستخدم من ٥ عناصر ثم يبحث عن أصغر قيم فيها ويطبع أصغر قيمة وثاني أصغر قيمة

```
#include<iostream.h>
main()
{ int x[5],i,j,min1,min2;
cout<<"enter the components of array\n";
for(i=0;i<5;i++)
cin>>x[i];
min1=min2=x[0];
for(i=0;i<5;i++)
if(x[i]<min1)
{min2=min1;
min1=x[i];}
cout<<"\n"<<min1<<"\t"<<min2;}
```

الشرح: يقوم هذا البرنامج بإدخال عناصر مصفوفة من ٥ عناصر ثم إعطاء قيم ابتدائية للمتغيرين min1 & min2 وهي تساوي قيمة أول عنصر في المصفوفة ثم يبحث عن أصغر عنصر في المصفوفة ويسنده للمتغير min1 بينما يحتفظ المتغير min2 بالقيمة قبل الأخيرة للمتغير min1 ثم يطبع القيم

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

وبعكس الإشارة في الشرط السابق من(if(x[i]<min1 الى (if(x[i]>min1 يمكن إيجاد أكبر قيمتين

--المصفوفات الرمزية

لإدخال كلمة وطباعتها لابد من الإعلان عن مصفوفة من نوع char ثم تحديد حجمها حيث size=number of letters +1 مثل ;char x[6]="mazen" حيث أن number of letters هو عدد حروف الكلمة بشرط أن لا تحتوي المصفوفة على فراغات فإذا احتوت على فراغ فإن ما بعد الفراغ يتم إهماله وسيتم معالجة مشكلة الفراغ لاحقاً في درس المؤشرات والسلاسل الحرفية وفي المثال السابق كان الإدخال مباشر بين علامتي تنصيص مزدوجة وتكون طباعتها بذكر اسمها دون تحديد حجمها كما يلي cout<<x;

*برنامج لإدخال الإسم الأول والثاني واللقب إدخال مباشر وطباعتها

```
#include<iostream.h>
main()
{ char name1[10],name2[10],lastname[10];
cout<<"enter the first name";
cin>>name1;
cout<<"\nenter the second name";
cin>>name2;
cout<<"\nenter the last name";
cin>>lastname;
cout<<"\nyour name is : "<<name1<<"  "<<name2<<"  "<<lastname; }
```

عند إدخال الاسم الأول"mazen" والاسم الثاني "abbass" والثالث "rawna" تكون شاشة المخرجات كما في الشكل التالي

```
mazen  abbass  rawna
```

انظر الملحق الأول فقرة رقم ٢ "تابع المصفوفات الرمزية" قبل الانتقال الى المصفوفات الثنائية

ب – المصفوفات الثنائية: تتكون من مجموعة صفوف ومجموعة أعمدة ويتم الإعلان عن المصفوفات ثنائية البعد كما يلي:

int x[3][3]; or int x[3][3]={12,3,4,54,7,83,86,11,65};

التعريف السابق كان بالإدخال المباشر مرة والأخرى بالإدخال عن طريق المستخدم حيث يتم تحديد عدد كل من الصفوف والأعمدة جوار اسم المصفوفة أما حجم المصفوفة فيساوي عدد الصفوف مضروبا في عدد الأعمدة

في المصفوفات الثنائية لابد أن تسبق كل عملية إدخال أو إخراج بدوارتين الأولى للأعمدة والثانية للصفوف مثل:

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

١-إدخال عناصر المصفوفة x التي تتكون من ٣ صفوف و٣ أعمدة

```
for(i=0;i<3;i++)
for(j=0;j<3;j++)
cin>>x[i][j];
```

٢- إخراج(طباعة) عناصر المصفوفة x التي تتكون من ٣ صفوف و٣ أعمدة

```
for(i=0;i<3;i++)
{for(j=0;j<3;j++)
{cout<<x[i][j]<<"\t";}
cout<<"\n";}
```

طبعاً أثناء الإخراج لابد من إجراء بعض التنسيقات متل "\t" أو "\n" أو الحاصرتين {}

*برنامج يقوم بقراءة مصفوفة ثنائية تتكون من ٣ صفوف و٣ أعمدة وطباعتها

```
#include<iostream.h>
main()
{ int x[3][3]={1,2,3,4,5,6,7,8,9},i,j;
for(i=0;i<3;i++)
{for(j=0;j<3;j++)
{cout<<x[i][j]<<"\t";}
cout<<"\n";} }
```

1	2	3
4	5	6
7	8	9

*برنامج لإدخال مصفوفة مكونة من ٤ صفوف و٥ أعمدة وطباعتها حيث أن عدد عناصر المصفوفة يساوي عدد الصفوف مضروباً في عدد الأعمدة أي ٤*٥=٢٠

```
#include<iostream.h>
main()
{int x[4][5],i,j;
for(i=0;i<4;i++)
for(j=0;j<5;j++)
cin>>x[i][j];
cout<<"\n\n";
for(i=0;i<4;i++)
{for(j=0;j<5;j++)
{cout<<x[i][j]<<"\t";}
cout<<"\n";}}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

عند تنفيذ البرنامج على المترجم وإدخال عناصر المصفوفة الأرقام من ١ الى ٢٠ تكون المخرجات كما في الشكل التالي

1	2	3	4	5
6	7	8	9	10
11	12	13	14	15
16	17	18	19	20

خصائص المصفوفة الثنائية التي عدد الصفوف i فيها مساوياً لعدد الأعمدة j أي المصفوفة المربعة index =i=j حيث index هو عدد الصفوف أو عدد الأعمدة

١- القطر الرئيسي: رقم الصف يساوي رقم العمود (i==j)
٢- القطر الثانوي: (i+j==index-1)
٣- ما فوق القطر الرئيسي رقم الصف أصغر من رقم العمود (i<j)
٤- ما تحت القطر الرئيسي رقم الصف أكبر من رقم العمود (i>j)
٥- ما فوق القطر الثانوي (i+j<=index-2)
٦- ما تحت القطر الثانوي (i+j>=index)

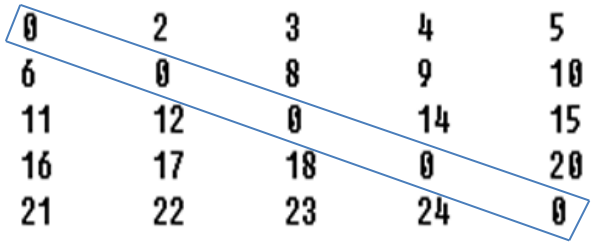
أمثلة على المصفوفة المربعة :

*برنامج لقراءة مصفوفة من ٥ صفوف و٥ أعمدة (٢٥ عنصر) وطباعتها وتصفير القطر الرئيسي أي أن الشرط if(i==j)

```
#include<iostream.h>
main()
{int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{if(i==j)
cout<<"0"<<"\t";
else cout<<x[i][j]<<"\t";}
cout<<"\n";}}
```

عند تنفيذ البرنامج تكون عناصر القطر الرئيسي كلها أصفار كما في الشكل التالي:

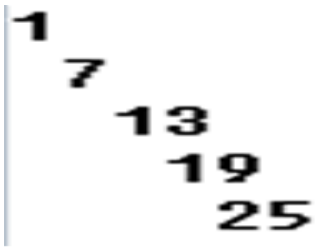
مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com



*برنامج لطباعة القطر الرئيسي فقط

```
#include<iostream.h>
main()
{int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
 {for(j=0;j<5;j++)
{if(i==j)
cout<<x[i][j];
else cout<<" ";}
cout<<"\n";} }
```

عند تنفيذ الكود السابق على المترجم تكون المخرجات هي عناصر القطر الرئيسي كما في الشكل التالي



*برنامج لتصفير ما فوق القطر الرئيسي أي أن الشرط(if(i<j

```
#include<iostream.h>
main()
{ int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{ if(i<j)
cout<<"0"<<"\t";
else cout<<x[i][j]<<"\t";}
cout<<"\n";}}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

عند تنفيذ البرنامج على المترجم يتم تصفير كل القيم الواقعة فوق القطر الرئيسي وتكون المخرجات كما في الشكل التالي

1	0	0	0	0
6	7	0	0	0
11	12	13	0	0
16	17	18	19	0
21	22	23	24	25

*برنامج لتصفير عناصر ما تحت القطر الرئيسي أي أن الشرط if(i>j)

```
#include<iostream.h>
main()
{int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{if(i>j)
cout<<"0"<<"\t";
else cout<<x[i][j]<<"\t";}
cout<<"\n";} }
```

عند تنفيذ الكود السابق يتم تصفير كل العناصر الواقعة تحت القطر الرئيسي وتكون المخرجات كما في الشكل التالي:

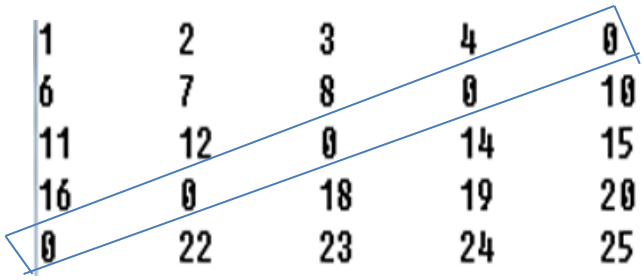
1	2	3	4	5
0	7	8	9	10
0	0	13	14	15
0	0	0	19	20
0	0	0	0	25

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

برنامج لتصفير القطر الثانوي وطباعة بقية عناصر المصفوفة كما هي:

```
#include<iostream.h>
main()
{ int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{ if(i+j==4)
cout<<"0"<<"\t";
else cout<<x[i][j]<<"\t";}
cout<<"\n";}}
```

عند تنفيذ الكود السابق على المترجم يتم تصفير عناصر القطر الثانوي وتكون مخرجات البرنامج كما في الشكل التالي



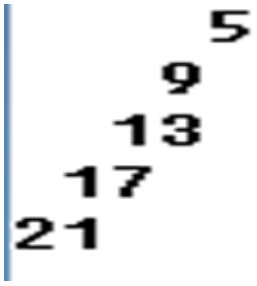
1	2	3	4	0
6	7	8	0	10
11	12	0	14	15
16	0	18	19	20
0	22	23	24	25

*برنامج لطباعة عناصر القطر الثانوي فقط أي أن الشرط if(i+j==index-1)

```
#include<iostream.h>
main()
{ int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{ if(i+j==4)
cout<<x[i][j];
else cout<<" "; }
cout<<"\n";}}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

عند تنفيذ الكود السابق على المترجم يتم طباعة عناصر القطر الثانوي فقط



*برنامج لتصفير عناصر ما فوق القطر الثانوي وطباعة بقية المصفوفة كما هي

```
#include<iostream.h>
main()
{ int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{ if(i+j<=3)
cout<<"0"<<"\t";
else cout<<x[i][j]<<"\t";}
cout<<"\n";}}
```

عند تنفيذ الكود على المترجم تكون المخرجات كما في الشكل التالي:

0	0	0	0	5
0	0	0	9	10
0	0	13	14	15
0	17	18	19	20
21	22	23	24	25

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج لتصفير عناصر ما تحت القطر الثانوي وطباعة بقية العناصر كما هي أي أن الشرط if(i+j>=index)

```
#include<iostream.h>
main()
{ int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{ if(i+j>=5)
cout<<"0"<<"\t";
else cout<<x[i][j]<<"\t";}
cout<<"\n";}}
```

وعند تنفيذ الكود على المترجم يتم تصفير عناصر ما تحت القطر الثانوي وتكون المخرجات كما في الشكل التالي

1	2	3	4	5
6	7	8	9	0
11	12	13	0	0
16	17	0	0	0
21	0	0	0	0

*برنامج لطباعة وجمع عناصر المصفوفة وطباعة ناتج الجمع

```
#include<iostream.h>
main()
{ int x[5][5]={1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25};
int i,j,y=0;
for(i=0;i<5;i++)
{for(j=0;j<5;j++)
{ y=y+x[i][j];
cout<<x[i][j]<<"\t";}
cout<<"\n";}
cout<<"\nthe sum is "<<y;}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

عند تنفيذ البرنامج السابق على المترجم تكون المخرجات كما في الشكل التالي

```
1       2       3       4       5
6       7       8       9       10
11      12      13      14      15
16      17      18      19      20
21      22      23      24      25

the sum is 325
```

*برنامج لإدخال مصفوفتين أحاديتين x[5] & y[5] حجم كل منهما ٥ عناصر ويدمجهما في مصفوفة أحادية ثالثة z[10] حجمها ١٠ عناصر ثم يطبع جميع المصفوفات

```
#include<iostream.h>
main()
{int x[5]={10,20,30,40,50},y[5]={60,70,80,90,100},z[10],i;
for(i=0;i<5;i++)
  {z[i]=x[i];
   z[i+5]=y[i];}
cout<<"the array x\n";
for(i=0;i<5;i++)
cout<<x[i]<<"\t";
cout<<"\nthe array y\n";
for(i=0;i<5;i++)
cout<<y[i]<<"\t";
cout<<"\nthe array z\n";
for(i=0;i<10;i++)
cout<<z[i]<<"\t";}
```

يقوم البرنامج السابق بتحويل مصفوفتين أحاديتين الى مصفوفة أحادية

عند تنفيذ الكود السابق تكون المخرجات كما يلي:

```
the array x
10      20      30      40      50
the array y
60      70      80      90      100
the array z
10      20      30      40      50      60      70      80      90      100
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج لتحويل مصفوفة أحادية z[10] تتكون من ١٠ عناصر الى مصفوفتين أحاديتين x[5] و y[5] تتكون كل منهما من ٥ عناصر

```
#include<iostream.h>
main()
{int z[10]={1,2,3,4,5,6,7,8,9,10},x[5],y[5],i;
for(i=0;i<5;i++)
{x[i]=z[i];
 y[i]=z[i+5];}
cout<<"the array z\n";
for(i=0;i<10;i++)
cout<<z[i]<<"\t";
cout<<"the array x\n";
for(i=0;i<5;i++)
cout<<x[i]<<"\t";
cout<<"\nthe array y\n";
for(i=0;i<5;i++)
cout<<y[i]<<"\t";}
```

عند تنفيذ الكود السابق على المترجم تكون المخرجات كما يلي

```
the array z
1       2       3       4       5       6       7       8       9       10
the array x
1       2       3       4       5
the array y
6       7       8       9       10
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

السجلات(structures):

نوع جديد من أنواع البيانات يقوم المستخدم بإنشائه مثله مثل int & float & char وغيرها ولكن السجلات (التراكيب) يمكن أن تحتوي على نوع واحد من الأنواع السابقة أو عدة أنواع أو حتى جميعها ويتم الإعلان عن السجل قبل الدالة الرئيسية main وذلك بإستخدام كلمة struct ثم تسمية السجل بإسم معين يخضع لقواعد التسمية المعروفة مسبقاً ويتم داخل السجل الإعلان عن المتغيرات دون إجراء أي عمليات حسابية أو منطقية والشكل العام للسجلات كما يلي

```
struct structure_name
        {first variable;
         second variable;};
```

مثال:

يتم الإعلان عن سجل (student) يحتوي على إسم الطالب من ٢٠ حرف ورقمه في الكشف ودرجات مادتين كما يلي:

```
struct student
      {char name[20];
       int number;
       float subject1,subject2}x;
```

أما الإعلان عن متغير(x) من نوع السجل فيتم بطريقتين الأولى: بذكر إسم المتغير قبل إنهاء السجل بعد إغلاق الحاصرة وقبل الفرزة المنقوطة كما في المثال السابق باللون الأحمر أما الطريقة الثانية : فتتم داخل الدالة main وذلك بذكر اسم السجل كنوع بياني ثم المتغير x كما يلي ;student x وفي كلتا الطريقتين نلاحظ أن المتغير x من نوع student وله إسم ورقم في الكشف ودرجة المادة الأولى ودرجة المادة الثانية

ولا نذكر اسم السجل داخل الدالة main الا عند الإعلان عن متغير من نوع السجل فقط أما إدخال بيانات المتغير x من نوع السجل student فيتم بذكر اسم المتغير ثم نقطة(.) ثم البيان المطلوب مثل الاسم أو الرقم كما يلي

```
cin>>x.name;
cin>>x.number;
```

وكذلك عملية الإخراج بنفس الطريقة

```
cout<<x.name;
cout<<x.number;
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

ويستفاد من السجلات عندما يكون المطلوب إدخال نفس البيانات لعدد كبير من الأشخاص حيث يتم الإعلان عن مصفوفة من نوع السجل حجمها هو عدد الأشخاص المطلوب إدخال بياناتهم
*برنامج يقوم بإدخال البيانات التالية (الاسم- درجة الرياضيات- درجة اللغة الإنجليزية – درجة البرمجة) ل٣٠ طالب ثم حساب المعدل الكلي لكل طالب وطباعة جميع البيانات لجميع الطلاب

```
#include<iostream.h>
struct std
      {char name[20];
       int sub1,sub2,sub3;
       float rate;};
main()
{std x[30];
for(int i=0;i<30;i++)
   {cout<<"\nenter the "<<i+1<<" student name :  ";
   cin>>x[i].name;
   cout<<"\nenter the "<<i+1<<" math deg :  ";
   cin>>x[i].sub1;
   cout<<"\nenter the "<<i+1<<" engli deg :  ";
   cin>>x[i].sub2;
   cout<<"\nenter the "<<i+1<<" prog deg :  ";
   cin>>x[i].sub3;
   x[i].rate=(x[i].sub1+x[i].sub2+x[i].sub3)/3;}
cout<<"name is    math deg    engli deg    prog deg    rate is\n";
for(i=0;i<30;i++)
{cout<<x[i].name<<"\t\t"<<x[i].sub1<<"\t\t"<<x[i].sub2<<"\t\t"
<<x[i].sub3<<"\t\t"<<x[i].rate<<"\n";}}
```

الشرح: قمت بالإعلان عن مصفوفة الطلاب X من نوع السجل std حجمها هو ٣٠ أي عدد الطلاب ثم بإستخدام الدوارة for قمت بإدخال بيانات كل طالب

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*تمرين :- برنامج نظام بنك ل ٥٠ زبون بحيث تكون بيانات الزبون هي (الاسم الأول– رقم الهاتف – رصيده الكلي- الرصيد المسحوب-الرصيد المتبقي)

```
#include<iostream.h>
struct clnt
   {char name[20],tel[15];
    long int account,taked,rem;};
void main()
{clnt x[50];
int i,j;
for(i=0;i<50;i++)
    {j=i+1;
    cout<<"enter the name of "<<j<<" client : ";
    cin>>x[i].name;
    cout<<"enter the telephone of "<<j<<" client : ";
    cin>>x[i].tel;
    cout<<"enter the money of "<<j<<" client : ";
    cin>>x[i].account;
    cout<<"enter the taked money of "<<j<<" client : ";
    cin>>x[i].taked;
    x[i].rem=x[i].account-x[i].taked;}
cout<<"name\t  tel\t     money\t  taked\t  remained\n";
for(i=0;i<50;i++)
{ cout<<x[i].name<<"\t  "<<x[i].tel<<"\t  "<<x[i].account<<"\t
"<<x[i].taked<<"\t  "<<x[i].rem<<"\n";} }
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

-الدوال functions:-

أ-الدوال المكتبية القياسية Standard Library Functions

هي الدوال المبنية داخلياً(built-in functions) التي تأتي مع مترجم لغة c++ وهي كثيرة جداً وتقع كل مجموعة منها داخل مكتبة محددة مثل

اسم المكتبة	استخدامها
iostream	لتعريف دوال الإدخال والإخراج input & output functions
conio	لتعريف دوال نظام dos
math	لتعريف دوال الرياضيات
string	لتعريف دوال السلاسل الحرفية
stdio	لتعريف دوال الإدخال والإخرج الخاصة بلغة c
graphics	لتعريف دوال الرسومات
time	لتعريف دوال الوقت

وغيرها من المكتبات أما كيفية استدعاء الدوال المكتبية داخل البرنامج فيتم كما يلي:

١- تحديد وتضمين الملف الذي يحتوي على الدالة المستخدمة ويتم ذلك من خلال الأمر (#include)

٢- استدعاء الدالة ويتم ذلك بكتابة اسم الدالة الصريح والذي يجب أن يطابق الاسم الموجود داخل الملف الذي يحوي هذه الدالة

٣- معرفة عدد المعلمات أو الوسائط (parameters) لكل دالة.

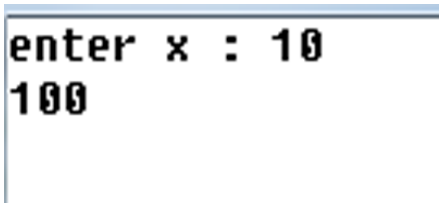
٤- نوع القيمة التي تعيدها الدالة

*برنامج يقوم برفع العدد x لأس ٢ أي يقوم بإيجاد مربع العدد x باستخدام الدالة pow التي تقع في المكتبة math

```
#include<iostream.h>
#include<math.h>
main()
{int x;
cout<<"enter x : ";
cin>>x;
cout<<pow(x,2);}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

عند تنفيذ الكود السابق وإدخال الرقم ١٠ للمتغير x تكون المخرجات كما يلي:

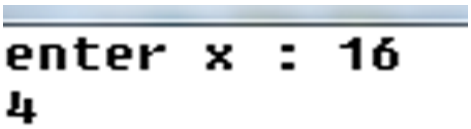


الشرح: أولاً قمت بإدخال قيمة x ثم استخدمت الدالة pow(x,2) لرفع المتغير x الى الرتبة الثانية فالدالة pow تستقبل متغيرين pow(x,y) حيث y هو الرتبة المطلوبة وبنفس الطريقة يمكن إيجاد المكعب

*برنامج لإدخال عدد x ثم يقوم بإيجاد الجذر التربيعي له باستخدام الدالة sqrt

```
#include<iostream.h>
#include<math.h>
main()
{int x;
cout<<"enter x : ";
cin>>x;
cout<<sqrt(x);}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي



ب- دوال المستخدم User-defined functions:

هي الدوال التي يقوم المبرمج بكتابتها لإنجاز مهمة معينة. والدالة هنا عبارة عن مجموعة من التعليمات المتسلسلة التي تشكل بمجموعها الهدف من استخدام الدالة ويستفاد من الدوال في سهولة تتبع البرنامج في حالة وجود أخطاء وكذلك يكون البرنامج أكثر ترتيباً كما يفيد استخدام الدوال في تقليل عدد أسطر البرنامج أما كتابة الدالة في لغة c++ فتأخذ الصيغة التالية:

```
Return_value_type  function_name (parameter list)
  {
     body of the function
     return value;
  }
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

حيث أن return_value_type هو نوع القيمة المرجعة وقد تكون القيمة المرجعة من نوع int أو float أو char أو غيرها وقد تكون الدالة لا تعيد قيمة فتكون من نوع void ولا يصح كتابة الجمل التعريفية وكذلك عمليات الإدخال والإخراج في الدوال التي ترجع قيم return values functions

و function_name هو اسم الدالة وفق قواعد تسمية المتغيرات

و parameter list هي قائمة بعدد المعلمات (المتغيرات) وأنواعها

و body of the function هو جسم الدالة أي التعليمات والأوامر التي تحدد عمل الدالة

و return value هي القيمة المرجعة وإذا كانت الدالة من نوع void فلا يوجد قيمة مرجعة ويتم استدعاء الدالة بذكر اسمها داخل الدالة الرئيسية أو أي دالة أخرى

أمثلة على الدوال المبنية عن طريق المستخدم

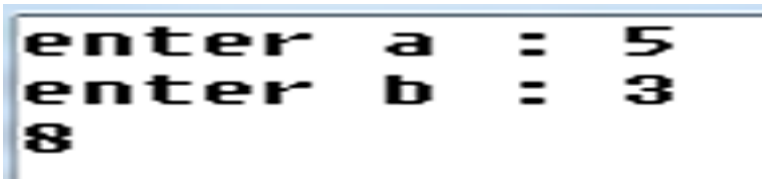
*برنامج يقوم بإدخال عددين x & y من النوع الصحيح عن طريق المستخدم ويطبع ناتج الجمع باستخدام الدوال

```
#include<iostream.h>
int sum(int x, int y)
    {int m;
     m=x+y;
    return m;}
main()
{int a,b;
cout<<"enter a : ";
cin>>a;
cout<<"enter b : ";
cin>>b;
cout<<sum(a,b);
```

الشرح: عرفنا دالة من نوع int لأنها ترجع قيمة من نوع int وأسميناها sum وهي تستقبل متغيرين من نوع int وتسند جمعهما لمتغير آخر من نفس النوع ثم ترجع الدالة ناتج الجمع أما داخل الدالة الرئيسية main فقد عرفنا متغيرين من نوع int كما في الدالة sum وأدخلنا قيمهما وأستدعينا الدالة sum لتنفذ عملها عليهما

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

عند إدخال الرقم ٥ للمتغير a و٣ للمتغير b تكون المخرجات كما يلي:



وبنفس الطريقة يمكن إجراء عملية الطرح والضرب والقسمة
*برنامج الحاسبة يقوم بالعمليات الأربع باستخدام الدوال ويسمح بالاستمرار في الاستخدام بقدر ما يريد المستخدم

```
#include<iostream.h>
int sum(int x,int y)
  {int m=x+y;
   return m;}
int sub(int x,int y)
  {int m=x-y;
   return m;}
int mul(int x,int y)
  {int m=x*y;
   return m;}
int div(int x,int y)
  {int m=x/y;
   return m;}
main()
{int a,b;
char c,d;
for(;;)
{cout<<"enter a : ";
cin>>a;
cout<<"enter the operation : ";
cin>>c;
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

```
cout<<"enter b : ";
cin>>b;
switch(c)
{  case'+':cout<<"the result is  "<<sum(a,b);break;
   case'-':cout<<"the result is  "<<sub(a,b);break;
   case'*':cout<<"the result is  "<<mul(a,b);break;
   case'/':cout<<"the result is  "<<div(a,b);break;
   default:cout<<"undefined operation";break;}
cout<<"\ndo you want to continue? (Y/N)";
cin>>d;
if(d=='Y')
{continue;
cout<<"\n";}
else {cout<<"\ngoodboy";break;}}}
```

*برنامج يسمح للمستخدم بإدخال عدد من النوع الكسري float وإذا كان الجزء الكسري أكبر من أو يساوي 0.5 فإن البرنامج يزيد الجزء الحقيقي بمقدار واحد وإذا كان الجزء الكسري أصغر من 0.5 يضل الجزء الحقيقي كما هو أي أن البرنامج يقوم بعملية التقريب فعند إدخال 7.5 أو 7.6 يطبع البرنامج الرقم 8 وعند إدخال 7.4 أو 7.3 يطبع الرقم 7 البرنامج باستخدام الدوال

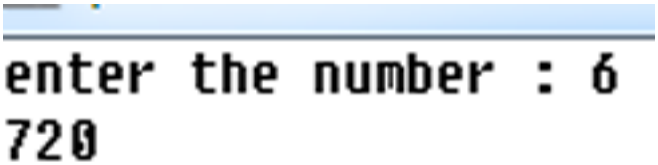
```
#include<iostream.h>
int func(float x)
  {int y=x;
   if(x-y>=0.5)
   return (y+1);
   else return y;}
main()
{float a;
cin>>a;
cout<<func(a);}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يقوم بإدخال عدد من النوع الصحيح ثم يقوم بإيجاد مضروبة حيث أن مضروب العدد 5 هو 1*2*3*4*5 ويساوي 120 باستخدام الدوال

```
#include<iostream.h>
long int fact(int x)
{int i;
long int y=1;
for(i=x;i>0;i--)
y=y*i;
return y;}
main()
{int a;
cout<<"enter the number : ";
cin>>a;
cout<<fact(a);}
```

الشرح: قمت بتعريف دالة من نوع long int حتى تعيد أرقام كبيرة وهي تستقبل عدد واحد x ثم عرفنا متغير y=1 ومتغير i=x ثم جعلنا دوارة تبدأ من i=x وتنتهي عند i=1 وتتناقص بمقدار ١ وفي كل دورة y=y*i حيث أن كل من i & y تتغير قيمته في كل دورة وبالتالي يتم ايجاد المضروب وعند إدخال العدد ٦ للمتغير x تكون المخرجات كما يلي



لأن 6*5*4*3*2*1=720

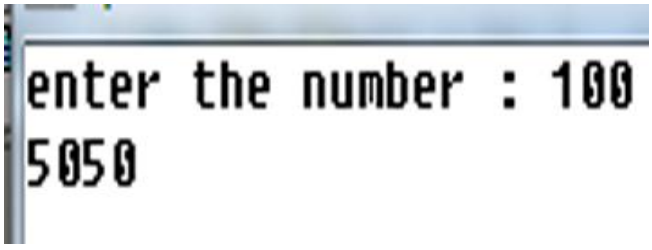
email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يقوم بإدخال عدد x من النوع الصحيح int ثم يطبع مجموع الأعداد من 1 الى x مثل عند ادخال العدد 5 يطبع 15 لأن 1+2+3+4+5=15 باستخدام الدوال

```
#include<iostream.h>
int summation(int x)
{int y=0,i;
for(i=1;i<=x;i++)
y=y+i;
return y;}
main()
{int a;
cout<<"enter the number : ";
cin>>a;
cout<<summation(a);}
```

الشرح : الفكرة في برنامج المجموع هي نفس فكرة المضروب فقط قمت بتغيير قيمة المتغير y=0 بدلاً من y=1 لأن في الجمع المحايد الجمعي هو 0 والمحايد الضربي هو 1

عند إدخال العدد ١٠٠ للمتغير a تكون المخرجات كما يلي

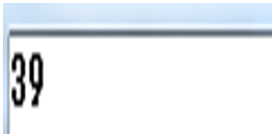


لأن 1+2+3+4+5+,,,,,,,+100=5050

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج باستخدام الدوال يقوم بإدخال عددين x & y ثم يجمع الأعداد من x الى y ويطبع ناتج الجمع مثلاً عند ادخال العددين (4,9) يطبع الرقم 39 لأن 4+5+6+7+8+9=39

```
#include<iostream.h>
int some_from_to(int x,int y)
        {int i,z=0;
         for(i=x;i<=y;i++)
         z=z+i;
         return z;}
main()
{int a=4,b=9;
cout<<some_from_to(a,b);}
```



عند تنفيذ الكود السابق تكون المخرجات كما يلي

***برنامج يبحث عن العامل المشترك الأكبر(g_c_d) بين رقمين x & y ثم يطبعة مثل عند ادخال العددين (40,50) يطبع الرقم ١٠ لأنه العامل المشترك الأكبر**

```
#include<iostream.h>
int g_c_d(int x,int y)
   {int i;
   for(i=x;i>0;i--)
     if(x%i==0&&y%i==0)
     break;
   return i;}
main()
{int a=40,b=50;
cout<<g_c_d(a,b);}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

العدد الأولي هو: العدد الذي لا يقبل القسمة إلا على نفسه أو على الواحد
*برنامج يسمح بإدخال عدد ثم يختبر إذا كان أولي يطبع "1" وإذا كان غير أولي يطبع "0"

```
#include<iostream.h>
int prime(int x)
  {if(x== 2 || x==3 || x==5 || x==7)
   return 1;
  else if(x%2==0 || x%3==0 || x%5==0 || x%7==0)
  return 0;
  else return 1;}
main()
{int i,a;
for(i=0;i<10;i++)
{cin>>a;
cout<<prime(a);
cout<<"\n";}}
```

*برنامج يقوم بإدخال عددين x & y ثم يبحث عن عامل مشترك ويختصرهما مثل عند ادخال العددين (200/600) يطبع العدد (1/3) لأن ٢٠٠÷٢٠٠=١ و٦٠٠÷٢٠٠=٣

```
#include<iostream.h>
void reduce(int x,int y)
  {int i;
   for(i=x;i>0;i--)
   {if(x%i==0 && y%i==0)
   break;}
   cout<<x/i<<"/"<<y/i;}
main()
{int a,b;
cout<<"enter a : ";
cin>>a;
cout<<"enter b : ";
cin>>b;
reduce(a,b);}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يسمح بإدخال رقم من ١ الى ٩ ثم يطبع اسم الكوكب في المجموعة الشمسية مرتباً من الأقرب الى الشمس باستخدام الدوال

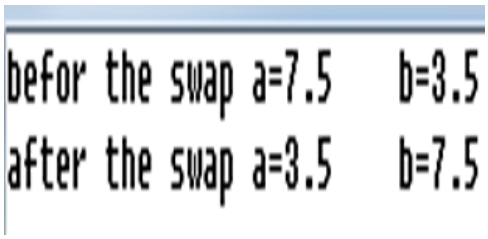
```
#include<iostream.h>
void func(int x)
   {switch(x)
    {case 1: cout<<"it is a Mercury\n";break;
     case 2: cout<<"it is a Venus\n";break;
     case 3: cout<<"it is a Earth\n";break;
     case 4: cout<<"it is a Mars\n";break;
     case 5: cout<<"it is a Jupiter\n";break;
     case 6: cout<<"it is a Saturn\n";break;
     case 7: cout<<"it is a Uranus\n";break;
     case 8: cout<<"it is a Neptune\n";break;
     case 9: cout<<"it is a Pluto\n";break;
     default: cout<<"error\n";} }
main()
{int a,i;
for(i=0;i<9;i++)
 {cout<<"enter the number a planet : ";
 cin>>a;
 func(a);}}
```

*برنامج يقوم بإدخال عددين a & b من النوع float ثم يبادلهما بالقيم باستخدام الدوال

```
#include<iostream.h>
void swap(float x,float y)
   {float z;
    z=x;
    x=y;
    y=z;
   cout<<"a="<<x;
   cout<<"\tb="<<y;}
main()
{float a=7.5,b=3.5;
cout<<"befor the swap a="<<a<<"\tb="<<b;
cout<<"\nafter the swap ";
swap(a,b);}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

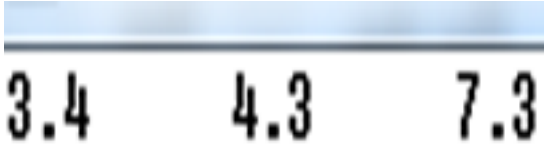
الشرح: قمت بتعريف المتغير z وأسندت له قيمة x القديمة ليحتفظ بها ثم أسندت قيمة y للمتغير x حيث x=y ثم نقلت قيمة x القديمة للمتغير y حيث y=z وهكذا تمت عملية المبادلة باستخدام متغير إضافي وعند تنفيذ الكود السابق تكون المخرجات كما يلي



*برنامج يقوم بإدخال ٣ قيم من نوع float هي i & j & k ثم يرتبها تصاعدياً باستخدام الدوال

```
#include<iostream.h>
void sort3(float x,float y, float z)
        {float a,b,c;
         if(x>y)
         {a=x;b=y;}
         else {a=y;b=x;}
        if(a>z)
        {c=a;a=z;z=c;}
        if(a>b)
        {y=a;x=b;}
        else {y=b;x=a;}
        cout<<x<<"\t"<<y<<"\t"<<z;}
main()
{float i=3.4,j=7.3,k=4.3;
sort3(i,j,k);}
```

عند تنفيذ الكود على المترجم تكون المخرجات كما يلي

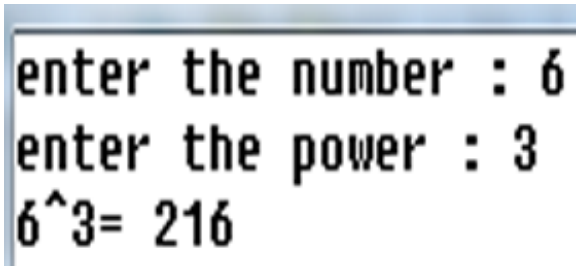


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

* تقوم الدالة المكتبية pow التي تقع ضمن المكتبة <math.h> برفع درجة أي عدد فمثلاً x=pow(2,5) تكون قيمة x=32 لأن
٢*٢*٢*٢*٢=٣٢ أي ٢ أس ٥ يساوي ٣٢ وكذلك pow(6,2)=36 لأن
٦*٦=٣٦ وكذلك pow(4,3)=64 لأن ٤ أس ٣ يساوي ٦٤
المطلوب: إكتب الدالة power(x,y) التي تقوم بعمل الدالة pow

```
#include<iostream.h>
int power(int x,int y)
    {int z=1,i;
     for(i=0;i<y;i++)
     z=z*x;
     return z;}
main()
{int a,b;
cout<<"enter the number : ";
cin>>a;
cout<<"enter the power : ";
cin>>b;
cout<<a<<"^"<<b<<"= "<<power(a,b);}
```

عند تنفيذ الكود السابق على المترجم وإدخال ٦ للأساس و ٣ للأس (القوة) تكون المخرجات كما يلي



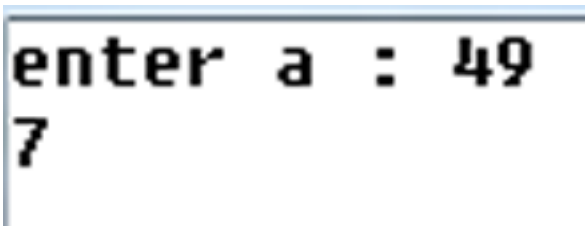
لأن ٦*٦*٦=٢١٦

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*تقوم الدالة المكتبية sqrt(x) التي تقع ضمن المكتبة <math.h> بإيجاد الجذر التربيعي للمتغير (x) اكتب الدالة func التي تقوم بنفس العمل

```
#include<iostream.h>
#include<math.h>
int func(int x)
    {int y;
     y=pow(x,0.5);
return y;}
main()
{int a;
cout<<"enter a : ";
cin>>a;
cout<<func(a);}
```

عند تنفيذ الكود السابق وإدخال الرقم ٤٩ تكون المخرجات كما يلي



لأن الجذر التربيعي ل ٤٩=٧

*اكتب الدالة root التي تقوم بإدخال عددين x & y من نوع int ثم تقوم بإيجاد الجذر y للرقم x مثل root(16,2) سيطبع البرنامج الرقم ٤ لأن الجذر التربيعي للعدد ١٦ =٤

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

```
#include<iostream.h>
int power(float x, float y)
      {float i,z=1;
      for(i=y;i>0;i--)
      z=z*x;
      return z;}
float root(float x, float y)
      {float i;
       for(i=x;i>0;i--)
       if(x==power(i,y))
       break;
       return i;}

main()
{ float a,b;
cout<<"enter the number : ";
cin>>a;
cout<<"enter the deg of root : ";
cin>>b;
cout<<root(a,b);}
```

عند ادخال a=64 و b=6 تكون المخرجات كما يلي:

```
enter the number : 64
enter the deg of root : 6
2
```

لأن الجذر السادس للعدد ٦٤ يساوي ٢ لأن ٢ أس ٦ =٦٤

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

-أمثلة متنوعة حول ما سبق :-

*برنامج يقوم بإدخال رقم ثم يطبع مربع من النجمات stars أبعاده هي الرقم المدخل فإذا أدخلت الرقم ٤ يكون عدد النجمات في الطول ٤ وفي العرض ٤

```
#include<iostream.h>
main()
{int x,i,j;
cout<<"enter the size : ";
cin>>x;
for(i=0;i<x;i++)
 {for(j=0;j<x;j++)
   {if(i==0 || i==x-1 || j==0 || j==x-1)
    cout<<"*";
else cout<<" ";}
cout<<"\n";}}
```

الشرح : أولاً قمت بإدخال قيمة x وقمت بكتابة دوارتين for حدهما الأعلى هو x-1 أما الشرط فهو إذا كان الصف الأول أو الصف الأخير أو العمود الأول أو العمود الأخير إطبع * وإلا إطبع فراغ space وعند إدخال الرقم ٥ تكون المخرجات كما يلي:

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يقوم بقراءة مصفوفة x[5] من نوع float حجمها ٥ عناصر ثم يعكس ترتيب المصفوفة بحيث تكون المصفوفة

5.8	2.6	9.0	3.4	7.1

وبعد إعادة الترتيب تكون المصفوفة

7.1	3.4	9.0	2.6	5.8

```
#include<iostream.h>
main()
{float x[5]={5.8,2.6,9.0,3.4,7.1};
int i;
for(i=0;i<5;i++)
cout<<x[i]<<"\t";
cout<<"\n";
for(i=4;i>=0;i--)
cout<<x[i]<<"\t";}
```

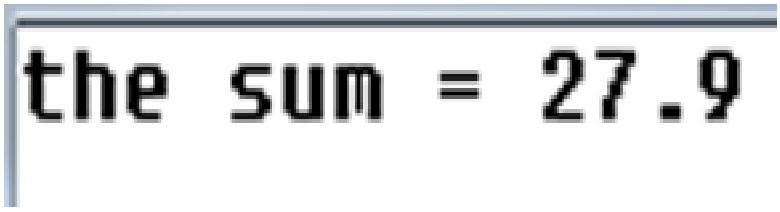
برنامج لقراءة المصفوفة

5.8	2.6	9.0	3.4	7.1

ثم يجمع عناصرها ويطبع ناتج الجمع

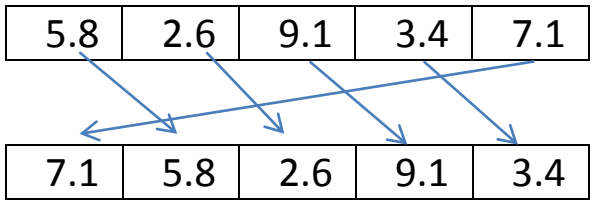
```
#include<iostream.h>
main()
{float x[5]={5.8,2.6,9.0,3.4,7.1},y=0;
int i;
for(i=0;i<5;i++)
y=y+x[i];
cout<<"the sum = "<<y;}
```

عند تنفيذ الكود السابق على المترجم تكون المخرجات كما يلي



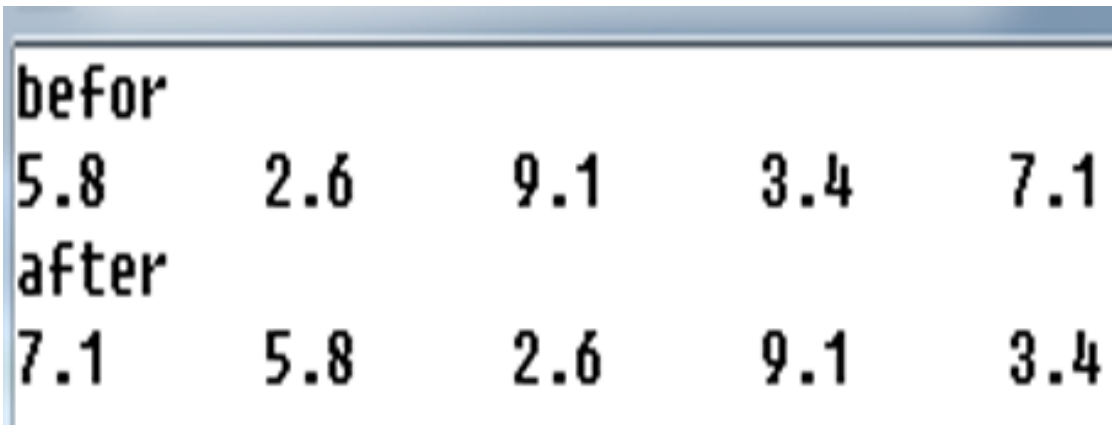
email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج لقراءة مصفوفة x[5] من نوع float تتكون من٥ عناصر ويقوم بعمل إزاحة من اليسار الى اليمين بمقدار خطوة واحدة بحيث تكون المصفوفة التالية كما يلي




```
#include<iostream.h>
main()
{float x[5]={5.8,2.6,9.1,3.4,7.1},y[5];
int  i;
y[0]=x[4];
for(i=1;i<5;i++)
y[i]=x[i-1];
cout<<"befor\n";
for(i=0;i<5;i++)
cout<<x[i]<<"\t";
cout<<"\nafter\n";
for(i=0;i<5;i++)
cout<<y[i]<<"\t";}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي

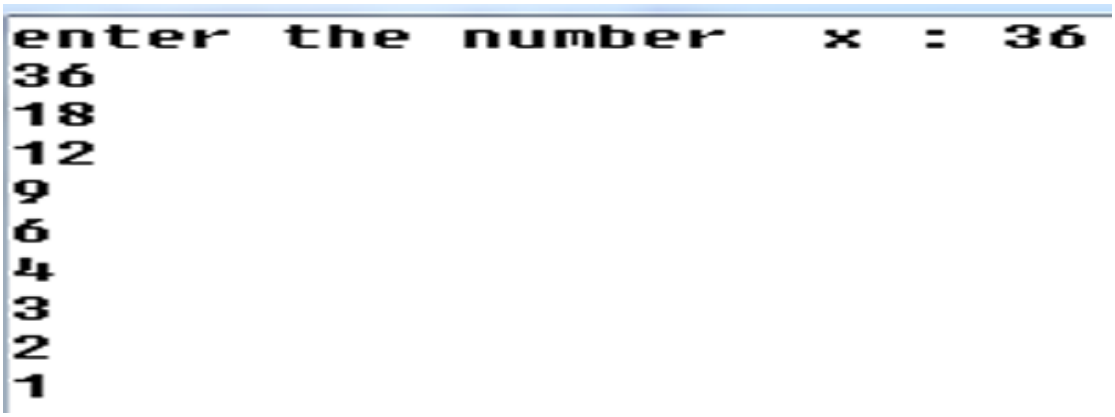


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يسمح بإدخال عدد x ثم يطبع جميع الأعداد التي x يقبل القسمة عليها مثلاً عند إدخال الرقم ١٢ سوف يطبع الأرقام (١٢,٦,٤,٣,٢,١) لأن الرقم ١٢ يقبل القسمة على جميع هذه الأرقام

```
#include<iostream.h>
main()
{int x,i;
cout<<"enter the number  x : ";
cin>>x;
for(i=x;i>0;i--)
if(x%i==0)
cout<<i<<"\n";}
```

عند تنفيذ الكود السابق على المترجم وإدخال الرقم 36 للمتغير x تكون المخرجات كما يلي



*برنامج يقوم بقراءة مصفوفة من الأعداد الصحيحة int ثم يبحث عن القيم المتكررة ويحذفها ويبقي أحدها فقط (أي هذا البرنامج لإزالة التكرار)

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

```
#include<iostream.h>
main()
{int x[10],i,j;
cout<<"enter the values : \n";
for(i=0;i<10;i++)
cin>>x[i];
cout<<"the values before eliminate repetitive values are \n";
for(i=0;i<10;i++)
cout<<x[i]<<"\t";
cout<<"\n";
for(i=0;i<10;i++)
for(j=0;j<10;j++)
if(x[i]==x[j] && i!=j)
x[j]=0;
cout<<"\nthe values after the eliminate \n";
for(i=0;i<10;i++)
if(x[i]!=0)
cout<<x[i]<<"\t";
else cout<<"\t";}
```

عند ادخال المصفوفة التالية الى البرنامج

```
10      20      30      10      40      50      10      60      60      70
```

يقوم البرنامج بحذف القيم التي تكررت وهي ١٠ و ٦٠ وتكون المخرجات كما يلي

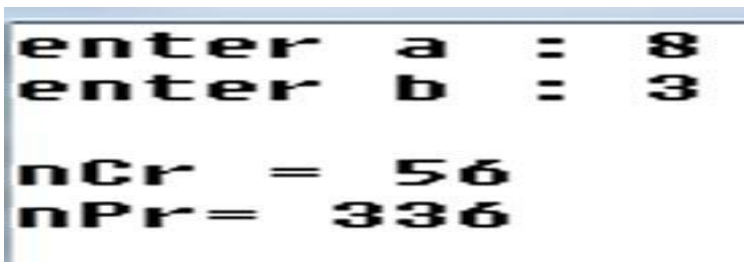
```
the values before eliminate repetitive values are
10      20      30      10      40      50      10      60      60      70


the values after the eliminate
10      20      30              40      50              60              70
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برنامج يقوم بإيجاد التوافيق nCr والتباديل nPr باستخدام الدوال

```
#include<iostream.h>
long int fact(int  x)
  {long int i,y=1;
  for(i=x;i>0;i--)
  y=y*i;
  return y;}
long int nCr(int n,int r)
  {long int m;
m=fact(n)/(fact(n-r)*fact(r));
return m;}
long int nPr(int n,int r)
  {long int m;
  m=fact(n)/fact(n-r);
return m;}
main()
{int a,b;
cout<<"enter a : ";
cin>>a;
cout<<"enter b : ";
cin>>b;
cout<<"\nnCr = "<<nCr(a,b);
cout<<"\nnPr= "<<nPr(a,b);}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي



email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

***برنامج يقوم بإدخال عدد x من نوع float ثم يقوم بتقريب هذا العدد (طريقة أسهل)**

```
#include<iostream.h>
int fun(float x)
        {int y=x+0.5;
         return y;}
main()
{float a;
cin>>a;
cout<<fun(a);}
```

*برنامج يقوم بادخال عدد x من نوع int ثم يجمع الأعداد من ١ الى أن يصل مجموعها أكبر من أو يساوي x عندها يرجع آخر عدد جمعه باستخدام الدوال

```
#include<iostream.h>
int enough(int x)
         {int i,y=0;
         for(i=1;i<x;i++)
          {y=y+i;
          if(y>=x)
          break;}
          return i;}
main()
{int a;
cin>>a;
cout<<enough(a);}
```

عند ادخال العدد ٩ سيطبع البرنامج الرقم ٤ لأن ١+٢+٣+٤=١٠ >=٩
وعند ادخال العدد ٢١ سيطبع الرقم ٦ لأن ١+٢+٣+٤+٥+٦=٢١>=٢١

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*المؤشرات pointers:-

عندما يتم الإعلان عن أي متغير في لغة c++ فإن موقع(عنوان) داخل ذاكرة الحاسوب الرئيسية سوف يخصص له ولغة c++ تهتم كثيراً بتقليل استهلاك الذاكرة وسرعة الوصول الى البيانات المخزنة فيها لهذا وفرت هذه اللغة ميزة جديدة من مزاياها وهي استخدام المؤشرات في الوصول الى البيانات المخزنة من خلال عناوينها

والمؤشر هو عبارة عن متغير يشير الى عنوان (موقع) متغير آخر علماً بأنه لا يحتفظ بقيمة المتغير الذي يشير اليه بل بعنوانه

- الإعلان عن المؤشرات:

يتم الإعلان عن المؤشرات في لغة c++ باستخدام النجمة(*) كما يلي

```
Data_type *Pointer_name;
```

حيث أن Data_type : نوع المؤشر ويجب أن يكون نفس نوع البيانات التي يشير اليها علماً ان المؤشر يخزن موقع وليس قيمة والمؤشر أيضاً له موقع في الذاكرة يختلف عن الموقع الذي يشير اليه
و Pointer_name : اسم المؤشر الذي يشار به الى عنوان في الذاكرة
أمثلة عن الإعلان عن المؤشرات

```
1) int *x;   2)char *x;   3)float *x;   4)int *a,*b;
```

عندما نريد أن نطبع عنوان(موقع) المتغير x نكتب & قبل المتغير كما يلي

```
cout<<&x;
```

وإذا أعلنا عن المؤشر x* فإننا نسند له موقع المتغير a كما يلي

```
x=&a;
```

أو عند التعريف int *x=&a وبالتالي فإن المؤشر x يشير الى عنوان المتغير a

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

كذلك يمكن الإعلان عن المؤشر x* الذي يشير الى موقع العنصر a بطريقتين كما يلي

```
1- int a;
   int *x=&a;
2- int a;
    int *x;
    x=&a;
```

وكلتا الطريقتين صحيحة وتؤدي نفس الغرض ومن الأخطاء الشائعة عند الإعلان عن المؤشرات الإعلان التالي

```
int a;
int *x=a;
```

الاعلان خطأ لأن المؤشر لا يأخذ قيم وإنما مواقع (عناوين) وفي هذا الإعلان تم اعطاء المؤشر قيمة a وليس عنوانه ويكون الإعلان صحيح بالطريقتين السابقتين

ويمكن تغيير قيمة المتغير a من ١٠ الى ٢٠ باستخدام المؤشرات كما يلي

```
int a=10;
int *x=&a;
*x=20;
cout<<a;
```

تصبح قيمة a=20 حيث أن المؤشر كان يشير الى قيمة a وتساوي ١٠ وتم تغيير القيمة التي بداخل الموقع الذي يشير اليه x الى ٢٠ فتتغير قيمة a=10 الى a=20

ولابد من التأكيد على ان المؤشر(x) في الاعلان int *x=&a; يشير الى عنوان(موقع) المتغير a في الذاكرة بينما الموقع في الذاكرة هو الذي يحتفظ بقيمة المتغير a

*برنامج لادخال عدد x من نوع int ادخال مباشر ثم يطبع موقعه في الذاكرة

```
#include<iostream.h>
main()
{int x=5;
int *a=&x;
cout<<a;}
```

عند تنفيذ الكود السابق فان ما سيتم طباعته هو عنوان في الذاكرة
ومن الملاحظ أنه عند طباعة عنوان فإننا لا نستخدم المعامل(*)

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

أما اذا أردنا طباعة القيمة التي يشير اليها المؤشر a فان الطباعة ستكون باستخدام المعامل السابق حيث يتم استخدام المعامل(*) عند الاعلان عن المؤشر فقط واذا استخدمناه في البرنامج فان ما سيتم طباعته هو قيمة مثلا انظر الكود التالي:

```
#include<iostream.h>
main()
{int x=5;
int *a=&x;
cout<<*a;}
```

لم يتغير في الكود السابق الا اضافة (*) عند الطباعة ولكن المخرجات تكون هي الرقم (5)

*برنامج يقوم بقراءة قيمة المتغير a ثم يقوم بتغييرها باستخدام المؤشر x

```
#include<iostream.h>
main()
{int a=10;
int *x=&a;
cout<<a;
*x=20;
cout<<"\n"<<a;}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي

الشرح: أولاً قمنا بالإعلان عن المتغير a واسندنا له القيمة ١٠ ثم أعلنا عن مؤشر x يشير الى المتغير a ثم طبعنا القيمة الحالية للمتغير a ثم استبدلنا القيمة التي يشير اليها المؤشر x بقيمة جديدة هي ٢٠ وبالتالي تغيرت قيمة a وطبعنا القيمة الجديدة في سطر جديد

- العمليات على المؤشرات :- قد تجري على المؤشر بعض العمليات هي:

١- العمليات الحسابية : تجري على المؤشر عمليتي الجمع والطرح وناتج عملية الجمع أو الطرح يعتمد على نوع البيانات التي يشير اليها المؤشر فاذا كان لدينا ثلاثة مؤشرات متنوعة هي

```
char *ch;     ولنفترض أن هذا المؤشر يشير الى الموقع ١٠٠
int *in;      ولنفترض أن هذا المؤشر يشير الى الموقع٢٠٠
float *flo;   ولنفترض أن هذا المؤشر يشير الى الموقع ٣٠٠
```

واذا أجرينا العمليات (ch++; - in++; - flo++) فان المؤشر ch سيؤشر الى الموقع ١٠١ لأن النوع char يستهلك بايت واحد فقط من الذاكرة

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

أما المؤشر in فانه سيؤشر الى الموقع ٢٠٢ لأن النوع int يستهلك ٢ بايت
والمؤشر flo سيؤشر الى الموقع ٣٠٤ لأن النوع float يستهلك ٤ بايت في الذاكرة
ونفس الكلام يسري على معامل النقصان(--)

٢- عمليات الاسناد: إذا عرفنا المؤشرين x و y وقلنا أن x=&a; فان المؤشر x يشير الى عنوان المتغير a واذا قلنا y=x فهذا يعني أن كل من x و y يشيران الى الموقع نفسه

٣- عمليات المقارنة : يمكن استخدام المعاملات العلاقية مع المؤشرات وبالتأكيد عملية المقارنة يجب أن تكون بين مؤشرات تشير الى متغيرات مثل

```
int x=5,y=3;
int *p1=&x,p2=&y;
if(p1>p2)
cout<<"p1 is a larger then p2";
else if(p1<p2)
cout<<"p2 is a larger then p1";
else cout<<"p1 equal p2";
```

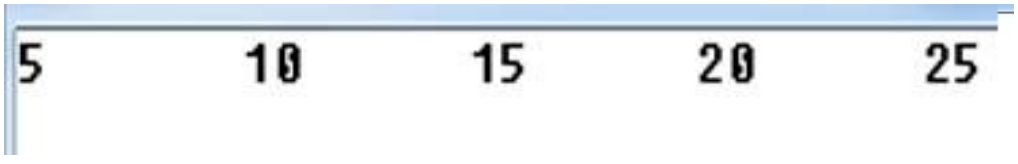
المعاملات التي بالخط الأحمر هي المقارنات بالإضافة الى علامة (==) و (=!)

- المؤشرات والمصفوفات :

أولاً : عزيزي القارئ قم بتنفيذ الكود التالي على المترجم ولاحظ ماذا سيحدث

```
#include<iostream.h>
main()
{int x[5]={5,10,15,20,25},i;
int *a;
a=x;
for(i=0;i<5;i++)
{cout<<*a<<"\t";
a++;}}
```

ان ما سيحدث هو طباعة عناصر المصفوفة وتكون المخرجات كما يلي



email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

الشرح:- قمت بالإعلان عن مصفوفة x[5] ومؤشر a* من نفس النوع ثم قلت أن a=x دون تحديد أي عنصر من x لأن a مؤشر وبهذه الطريقة فان a يؤشر الى موقع أول عنصر من x وكأني قلت a=&x[0] ثم داخل الدوارة for طبعت ما بداخل الموقع المخزون في المؤشر a وهو x[0]=5 ولان المؤشر والمصفوفة من نوع int فان الزيادة بمقدار ١أي(٢ مواقع) a++ تجعل المؤشر يشير الى ثاني عنصر في المصفوفة وهو x[1]=10 ثم واصل البرنامج على نفس المنوال الى ان اكتملت الطباعة

والمثال التالي يوضح طرق الزيادة في المؤشرات

```
#include<iostream.h>
main()
{int num[5],i,*p;
p=num;
*p=10; p++;
*p=20;
p=&num[2]; *p=30;
p=num+3; *p=40;
p=num; *(p+4)=50;
for(i=0;i<5;i++)
cout<<num[i]<<"\t";}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي:

```
10      20      30      40      50
```

الشرح: p=num أي أن المؤشر p يشير الى موقع العنصر الأول و p++ فانه يشير الى موقع العنصر الثاني و p=num[2] يشير الى العنصر الثالث p=num+3 فان المؤشر يشير الى العنصر الرابع و p=num تعيده الى العنصر الأول و p+4 أي موقع العنصر الخامس

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

- اكتب برنامج يسمح للمستخدم بتحديد حجم مصفوفة سلع تجارية ثم يسمح للمستخدم بإدخال ثمن كل سلعة ويضيف 5% الى الثمن كضريبة مبيعات ثم يطبع السعر الجديد؟
قد يبدأ البعض بالإعلان عن حجم المصفوفة وادخال بإحدى الطرق التالية

```
int i;
cin>>i;
int x[i];
```

حيث يبدأ بإدخال الحجم ثم يعلن عن المصفوفة وحجمها هو الرقم المدخل
لكن هذا الكود لن يتنفذ أبداً مهما حاولت...والبعض الآخر قد يقوم بتحديد حجم أقصى للمصفوفة ك ١٠٠٠ عنصر أو ٥٠٠٠ عنصر أو أكثر كما يلي:

```
int x[5000],i,j;
cin>>j;
for(i=0;i<j;i++)
cin>>x[i];
```

حيث j هو الحجم المدخل عن طريق المستخدم
صحيح أن هذا الكود سيتنفذ ويقوم بالعمل المطلوب ولمن عيوبه كثيرة مثل
أولاً : لابد من حد أعلى لحجم المصفوفة اذا تجاوزه الحجم المدخل أو وصل عدد السلع الى هذا الحجم فلن يستمر تنفيذ البرنامج
ثانياً : حجم البرنامج يزداد بازدياد عدد عناصر المصفوفة وقد يتطلب البرنامج مواصفات عالية لتشغيله والمبرمج الناجح يأخذ حجم البرنامج كقضية هامة ويعمل على كتابة برنامج يقوم بالعمل المطلوب وبأصغر حجم ممكن...
من الملاحظ ان الطرق السابقة وغيرها لا تفي بالغرض لذلك ومن فوائد المؤشرات أنها تسمح لك بتحديد حجم المصفوفة عن طريق المستخدم كما يلي:

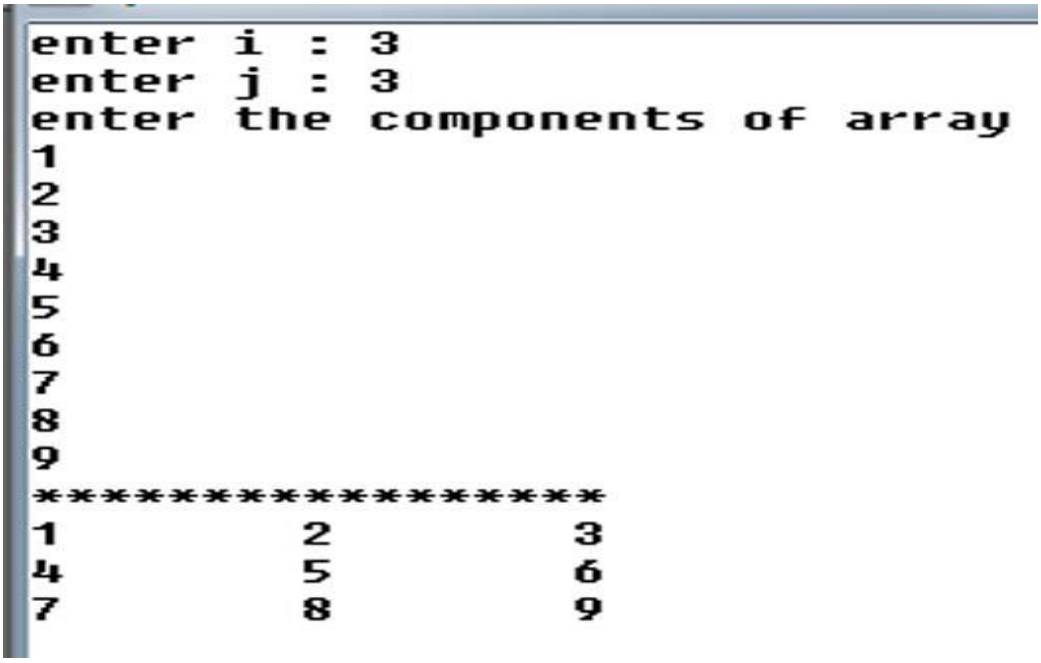
```
#include<iostream.h>
main()
{int i,j;
cout<<"enter the size of array : ";
cin>>i;
int *x=new int[i];
cout<<"the cost\t\tthe cost after adding 5%\n";
for(j=0;j<i;j++)
 {cin>>x[j];
  cout<<"\t\t\t"<<x[j]+(x[j]*0.05)<<"\n";}}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج يسمح للمستخدم بإدخال حجم مصفوفة ثنائية البعد بتحديد عدد الصفوف i وعدد الأعمدة j ثم يسمح للمستخدم بإدخال عناصر المصفوفة ويقوم بطباعة عناصر المصفوفة

```
#include<iostream.h>
main()
{int i,j,k,l,m;
cout<<"enter i : ";
cin>>i;
cout<<"enter j : ";
cin>>j;
int **x=new int*[i];
for(k=0;k<i;k++)
x[k]=new int[j];
cout<<"enter the components of array \n";
for(l=0;l<i;l++)
for(m=0;m<j;m++)
cin>>x[l][m];
cout<<"*****************\n";
for(l=0;l<i;l++)
{for(m=0;m<j;m++)
 {cout<<x[l][m]<<"\t";}
 cout<<"\n";}}
```

عند تنفيذ الكود السابق على المترجم وادخال حجم المصفوفة ٣*٣ وادخال عناصر المصفوفة الأرقام(١-٩) تكون المخرجات كما يلي :



email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

- المؤشرات والدوال:

يمكن استدعاء الدوال بطريقتين الطريقة الأولى الاستدعاء بالقيمة Call By Value وهي كما هو موضح في درس الدوال حيث كان الاستدعاء بالقيمة

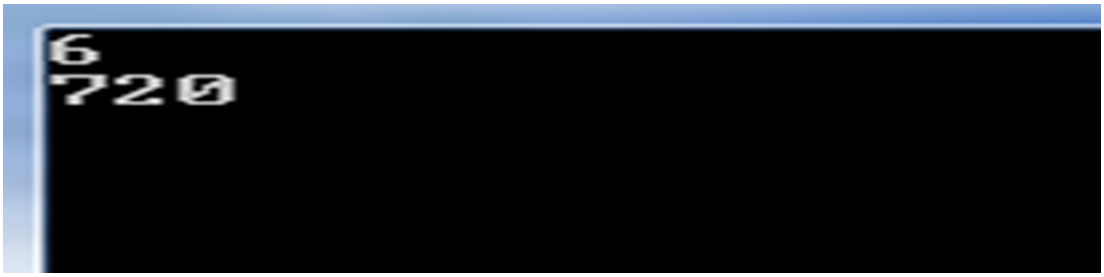
الطريقة الثانية هي الاستدعاء بالمرجع(العنوان) Call By Reference ونحن نعرف أن المؤشرات تشير الى عناوين المتغيرات والقيم

ويكون الاستدعاء بالعنوان من خلال الإعلان عن باراميترات الدالة من نوع المؤشرات أو المرجعيات pointers or references وعند استدعاء الدالة داخل الدالة الرئيسية main نرسل المرجع(العنوان) ولابد أن يكون المؤشر وكذلك العدد والمرجع المرسل من نفس النوع .....مثال

* اكتب برنامج باستخدام الدوال والمؤشرات يقوم باستقبال عدد من نوع long int ثم يوجد مضروب العدد

```
#include<iostream.h>
#include<conio.h>
long int fact(long int *p)
     {long int i,z=1;
      for(i=*p;i>0;i--)
      z=z*i;
      return z;}
main()
{long int a;
cin>>a;
cout<<fact(&a);
getch();}
```

عند تنفيذ الكود السابق على المترجم وادخال العدد ٦ تكون المخرجات كما يلي:



email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

ملاحظة: شاشة المخرجات سوداء لأن البيئة Borland c++

أما اذا قلنا في الاستدعاء cout<<fact(a); أو cout<<fact(*a); فلن يتم تنفيذ الكود لأن الدالة fact(*p) تستقبل عناوين (مرجعيات) ومؤشرات ولا تستقبل قيم ويمكن استدعائها بطريقة أخرى كما يلي:

```
Long int *x=&a;
cout<<fact(x);
```

ويمكن أن تكون باراميترات الدالة من نوع المرجعيات وليس المؤشرات كما يلي:

```
#include<iostream.h>
#include<conio.h>
int sum(int &a,int &b)
{int z=a+b;
return z;}
main()
{int i,j;
cin>>i>>j;
cout<<sum(i,j);
getch();}
```

وفي هذه الحالة فان الدالة تستقبل قيم ولا تستقبل مؤشرات أو مرجعيات وكذلك عملية الجمع داخل الدالة تنفذ بالقيم

*برنامج الحاسبة باستخدام المؤشرات والدوال تقوم بعملية الجمع والطرح والضرب والقسمة والتوافيق

```
#include<iostream.h>
int sum(int*a,int *b)
    {int c;
     c=*a+*b;
     return c;}
int sub(int *a,int *b)
    {int c;
     c=*a-*b;
     return c;}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

```
int mul(int *a,int *b)
      {int c;
      c=(*a)*(*b);
      return c;}
int div(int *a,int *b)
      {int c;
      c=(*a)/(*b);
      return c;}
int fact(int &a)
      {int i,c=1;
       for(i=a;i>0;i--)
       c=c*i;
       return c;}
int ncr(int &a,int &b)
       {int m;
       m=fact(a)/(fact(a-b)*fact(b));
       return m;}
main()
{int x,y;
char m;
cin>>x>>m>>y;
switch(m)
{case'+':cout<<x<<"+"<<y<<"="<<sum(&x,&y);break;
case'-':cout<<x<<"-"<<y<<"="<<sub(&x,&y);break;
case'*':cout<<x<<"*"<<y<<"="<<mul(&x,&y);break;
case'/':cout<<x<<"/"<<y<<"="<<div(&x,&y);break;
case'C':cout<<x<<"C"<<y<<"="<<ncr(x,y);break;
default: cout<<"error";break;}}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*السلاسل الرمزية في لغة c++ (strings in c++):-

الأنواع البيانية القياسية في لغة c++ هي : النوع الصحيح(int) والنوع الحقيقي (float) والنوع الرمزي (char) وليس هناك نوع السلاسل الرمزية، والسلاسل الرمزية هي كل نص تزيد حروف كلماته عن الرمز الواحد مع إعتبار أن رمز الفراغ (space) هو الفاصل بين هذه الكلمات

لكن الحاجة الماسة للتعامل مع السلاسل الرمزية ، أجبرت مستخدمي c++ على أن يوفروا طريقة غير مباشرة للتعامل مع السلاسل الرمزية ، والطريقة كانت باستخدام النوع الرمزي (char) من خلال الاعلان عن مصفوفة جميع عناصرها من النوع الرمزي أو من خلال مؤشر يشير الى النوع الرمزي (char) وباستخدام هذه الطريقة في التعامل مع السلاسل الرمزية نستطيع أن نعرف السلاسل الرمزية على أنها مجموعة من الرموز مرتبة بشكل متسلسل على أن تنتهي برمز الفراغ (\0).

- الإعلان عن السلاسل الرمزية:

هناك طريقتان للإعلان عن السلاسل الرمزية

١- استخدام المؤشرات : ويتم ذلك باستخدام مؤشر يشير الى عدد غير محدد من الرموز ضمن النوع (char) وطريقة الاعلان تكون كما يلي:

```
char *string _name;
```

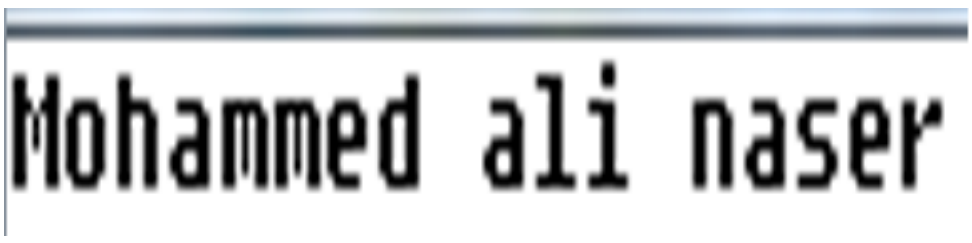
حيث (string_name) مؤشر يشير الى مجموعة من الرموز ويستخدم الرمز(*) عند الإعلان فقط ولا يستخدم عند الإدخال أو الطباعة وبهذه الطريقة يكون الإدخال مباشر أثناء الإعلان مثل

```
1)char *x="ALI";     2)char *x="mazen abbass";
```

وفي هذه الحالة تكون السلاسل الرمزية بين علامة التنصيص (الاقتباس) المزدوجة (" ") حتى لا يعتبرها المترجم أسماء متغيرات مثال

```
#include<iostream.h>
main()
{char *x="Mohammed ali naser";
cout<<x;}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي :



email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

أما إذا كان الإدخال غير مباشر (عن طريق المستخدم) فإن الإعلان عن السلاسل يتم بالطريقة التالية
1)char *x=new char;

وتسمى هذه الطريقة بالخزن الديناميكي وتتم باستخدام العامل (new) كما هو واضح

*برنامج يقوم بإدخال سلسلة رمزية عن طريق المستخدم وطباعتها

```
#include<iostream.h>
main()
{char *a=new char;
cout<<"please enter your first name : ";
cin>>a;
cout<<"your name is : "<<a;}
```

٢- استخدام المصفوفات : ويتم الإعلان عن السلاسل الرمزية باستخدام المصفوفات من خلال الإعلان عن مصفوفة جميع عناصرها من النوع الرمزي (char) بحيث كل عنصر من عناصرها يحتوي على رمز(حرف) واحد فقط ويتم الإعلان عن السلاسل كما يلي:

```
char array_name[size];
```

حيث array_name هو اسم المصفوفة التي عدد عناصرها size مثل char w[6]="mazen";

أي أن المصفوفة تتكون من ٦ عناصر ٥ منها حروف والعنصر الأخير خصص لرمز الفراغ ولأن الإدخال مباشر يمكن أن يكون الإعلان السابق كما يلي char w[]="mazen"; أي بدون تحديد حجم المصفوفة ولكن في حالة الإدخال المباشر فقط.
مثال: أي الإعلانات التالية صحيح وأيها خطأ ولماذا

```
a) char x[5]="naser";     b) char x[]="naser";     c) char x[6]="mohammed" ;
d) char x[6]="naser";     e) char *x[9]="mohammed";
f)char x[6]={'m','a','z','e','n'};   g)char x[100]="mazen";
h)char x[]={'m','a','z','e','n'};     i)char x[5]="ali";   j)char x="ali";
k)char x[]='ali';
```

a – خطأ: لأن حجم المصفوفة يجب أن يكون ٦ عناصر العنصر السادس لرمز الفراغ "\0"
b – صحيح :لأن حجم المصفوفة يتم تحديده تلقائياً
c – خطأ: لأن المصفوفة "mohammed" أكبر من الحجم المحدد لها عند الإعلان
d – صحيح
e – خطأ:لأننا أستخدمنا المصفوفات والمؤشرات معاً ويجب أن يكون
*x="mohammed"; أوx[9]="mohammed"

f- صحيح: لأنه لا مشكلة في الإدخال بهذه الطريقة إذا تم تحديد حجم المصفوفة

g – صحيح: لأن لا مشكلة إذا كان الحجم أكبر من المصفوفة المدخلة بل المشكلة هي العكس

h – خطأ: لأنه إذا لم يتم تحديد حجم المصفوفة فلا يصح إدخال المصفوفة حرفاً حرفاً

i – صحيح: لأن الحجم المحدد أكبر من المصفوفة المدخلة وبالتالي لا مشكلة

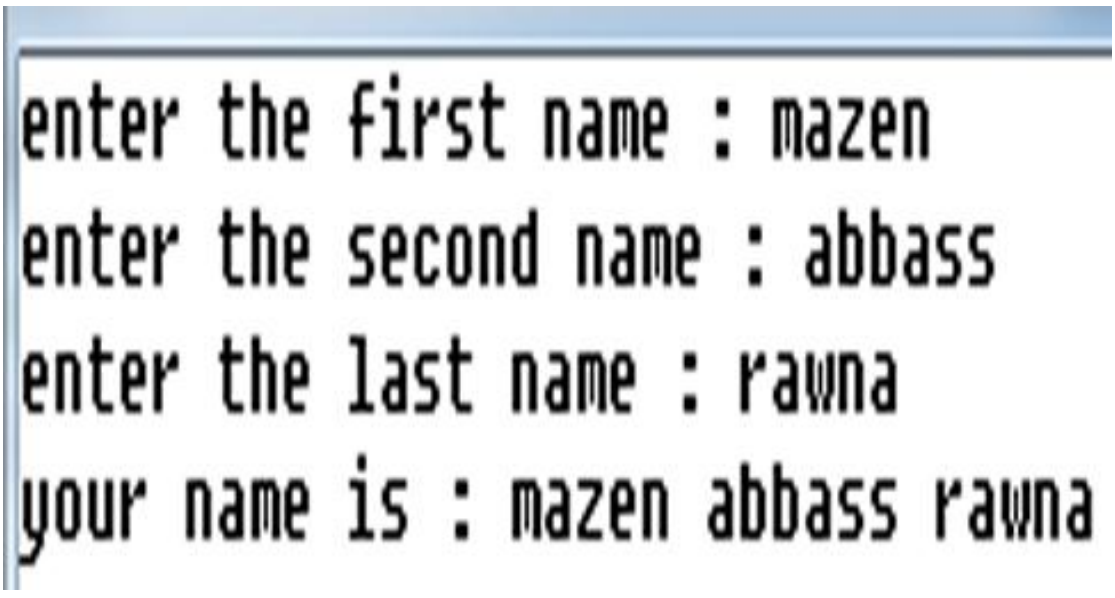
j – خطأ: لأن الإعلان عن حرف واحد فقط والمصفوفة ثلاثة حروف

k – خطأ : لأن علامة التنصيص(الاقتباس) مفردة ويجب أن تكون مزدوجة

*برنامج لإدخال الاسم الأول والثاني واللقب ثم طباعة الاسم كاملاً باستخدام المصفوفات

```
#include<iostream.h>
main()
{char x[10],y[10],z[10];
cout<<"enter the first name : ";
cin>>x;
cout<<"enter the second name : ";
cin>>y;
cout<<"enter the last name : ";
cin>>z;
cout<<"your name is : "<<x<<" "<<y<<" "<<z;}
```

عند الإدخال أو الإخراج لا نكتب حجم المصفوفة ونفعل ذلك فقط عند الإعلان وعند تنفيذ الكود السابق على المترجم وإدخال الاسم الأول "mazen" والثاني "abbass" والثالث rawna" تكون شاشة المخرجات كما يلي:



مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

أما عند كتابة البرنامج السابق باستخدام المؤشرات فيكون الكود كما يلي:

```
#include<iostream.h>
main()
{char *x=new char,*y=new char,*z=new char;
cout<<"enter the first name : ";
cin>>x;
cout<<"enter the second name : ";
cin>>y;
cout<<"enter the last name : ";
cin>>z;
cout<<"your name is : "<<x<<" "<<y<<" "<<z;}
```

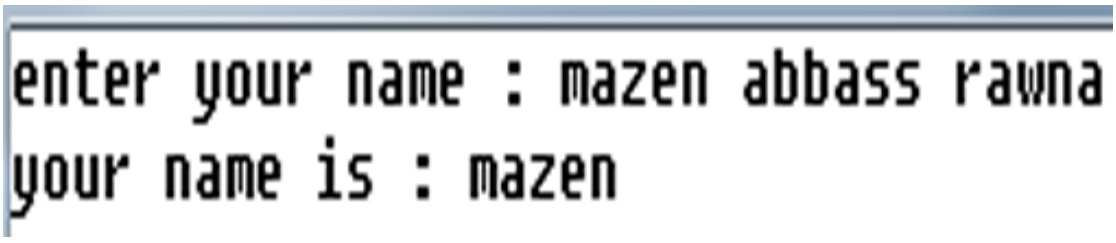
هذا البرنامج يقوم بنفس عمل البرنامج السابق وله نفس المخرجات

ولكن إذا أردنا إدخال الاسم الأول والثاني واللقب بمصفوفة واحدة حجمها كبير بما فيه الكفاية ماذا سيحدث؟؟

إن ما سيحدث هو أن المترجم سيقبل الاسم الأول فقط أما ما بعد رمز الفراغ "\0" فان المترجم لن يقبله أو يتعامل معه وذلك لأن رمز الفراغ يعتبر نهاية أي سلسلة حرفية والبرنامج التالي مثال على هذا *برنامج يقوم بإدخال الاسم الأول والثاني واللقب ثم يطبع الاسم كامل

```
#include<iostream.h>
main()
{char x[100];
cout<<"enter your name : ";
cin>>x;
cout<<"your name is : "<<x;}
```

عند تنفيذ الكود السابق وإدخال الإسم كاملاً تكون المخرجات كما يلي:



نلاحظ أن المترجم قبل الاسم الأول فقط ولم يقبل ما بعد الفراغ لذلك وفرت لغة c++ حلاً لهذه المشكلة (مشكلة الفراغ) وهي دالة cin.getline

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

الشكل العام للدالة cin.getline

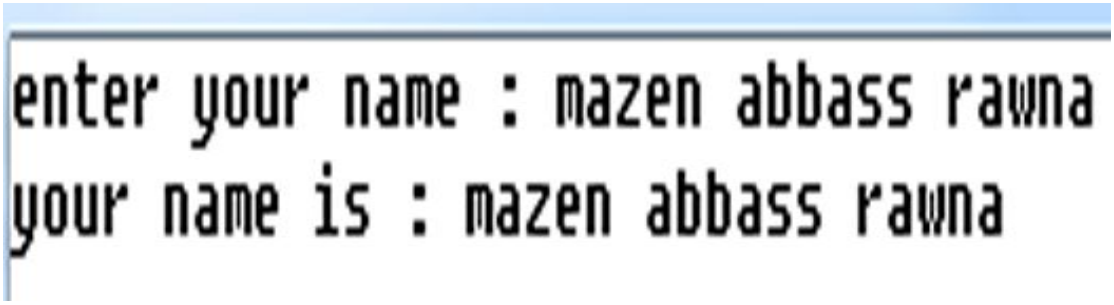
cin.getline(array_name,array_size)

حيث أن array_name هو اسم السلسلة أو المصفوفة دون تحديد حجمها و array_size هو عدد الرموز المدخلة وهذه الدالة تسمح بإدخال سلسلة رمزية تحتوي على الفراغات

*برنامج يقوم بإدخال الاسم الأول والثاني واللقب في مصفوفة واحدة باستخدام الدالة cin.getline ثم يطبع الاسم كامل

```
#include<iostream.h>
main()
{char x[100];
cout<<"enter your name : ";
cin.getline(x,100);
cout<<"your name is : "<<x;}
```

عند تنفيذ الكود السابق وإدخال الاسم الأول "mazen" والثاني "abbass" والثالث "rawna" تكون المخرجات كما يلي:



الدوال التي تتعامل مع السلاسل الرمزية:

الدالة	الوظيفة	ملاحظات عامة
strlen()	لايجاد عدد رموز السلسلة	لايجاد عدد رموز السلسلة x تستخدم كما يلي ;(strlen(x
strcat()	لدمج سلسلتين	ندمج x1 وx2 بشرط أن حجم x1 يستوعب السلسلتين معاً كمايلي strcat(x1,x2)
strcpy()	لنسخ سلسلة الى أخرى	ننسخ محتويات x1 الى x2 كما يلي ;strcpy(x2,x1)
strcmp()	للمقارنة بين محتويات سلسلتين	للمقارنة بين محتويات x1,x2 كما يلي strcmpy(x1,x2) تعيد قيمة موجبة إذا كان x1 أكبر وتعيد قيمة سالبة إذا كان x2 أكبر وتعيد "0" في حالة التساوي
strlwr()	**تحول عناصر سلسلة من الحروف الكبيرة الى الصغيرة**	لتحويل عناصر x الكبيرة الى صغيرة strlwr(x)
strupr()	**تحول عناصر سلسلة من الحروف الصغيرة الى الكبيرة**	لتحويل عناصر x من الصغيرة الى كبيرة strupr()

جميع هذه الدوال تقع ضمن المكتبة <string.h> لذلك لابد من تضمين هذه المكتبة في أي برنامج يستخدم هذه الدوال أو إحداها

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

١-الدالة (strlen): تستخدم لإيجاد عدد رموز(طول) سلسلة حرفية معينه على اعتبار ان الفراغ رمزاً:

*برنامج لإدخال سلسلة رمزية ويطبع طول السلسلة

```
#include<iostream.h>
#include<string.h>
main()
{char x[100];
cout<<"enter the string to find length :";
cin.getline(x,100);
cout<<strlen(x);}
```

*برنامج لإدخال كرت الخدش المكون من ١٤ رقم وإذا كان الرمز أطول أو أقصر من ١٤ يخبرنا بذلك

```
#include<iostream.h>
#include<string.h>
main()
{char x[20];
cout<<"please enter the number : ";
cin.getline(x,20);
if(strlen(x)>14)
cout<<"is larger then 14";
else if(strlen(x)<14)
cout<<"is smaller then 14";
else cout<<"true";}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*برنامج لإدخال كلمة السر إذا كانت "i love yemen" يطبع "true"
وإذا اختلف حرفاً واحداً أو فراغ أو كلمة خاطئة يطبع "error"

```
#include<iostream.h>
#include<string.h>
main()
{char x[100]="i love yemen",y[100];
int m,i;
cout<<"enter the pass : ";
cin.getline(y,100);
for(i=0;i<strlen(x);i++)
    {if(y[i]==x[i])
     {m=1;continue;}
      else {m=0;break;} }
if(m==1)
cout<<"true";
else cout<<"error";}
```

الشرح: قمت بإدخال جملة "i love yemen" إدخال مباشر ثم يصدر البرنامج أمراً بإدخال كلمة السر (السلسلة الرمزية) ثم جعلت الدوارة تبدأ من الصفر وتنتهي عند عدد رموز السلسلة باستخدام الدالة strlen(x) لأنها تمثل عدد صحيح وتقارن بين حروف كلمة السر والكلمة المدخلة حرفاً حرفاً وإذا وجد إختلاف يتوقف فوراً ويجعل قيمة m=0 وإذا لم يجد اختلاف يستمر بالدوران والمقارنة ويجعل قيمة m=1 وبعد الانتهاء يختبر قيمة m اذا كانت m=1 يطبع "true"وإذا كانت غير ذلك يطبع "error"

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

٢- الدالة strcat() : تستخدم لدمج سلسلتين رمزيتين مع بعض وذلك من خلال إضافة السلسلة الأولى الى الثانية وناتج عملية الدمج سيكون السلسلة الأولى لهذا يجب أن يكون عدد عناصر السلسلة الأولى يستوعب السلسلتين معاً
الشكل العام لهذه الدالة strcat(x,y);
حيث تدمج السلسلتين x & y ويجب أن يكون عدد عناصر السلسلة x يستوعب عدد عناصر السلسلتين معاً
*برنامج يقوم بقراءة سلسلتين ودمجهما معاً وطباعة كل منهما بعد الدمج

```
#include<iostream.h>
#include<string.h>
main()
{char x[20]="mazen ",y[20]="abbass";
strcat(x,y);
cout<<x;
cout<<"\n"<<y;}
```

عند تنفيذ الكود السابق على المترجم تكون المخرجات كما يلي:

```
mazen abbass
abbass
```

نلاحظ أن التغير حصل في المصفوفة x

٣- الدالة strcpy() : تستخدم هذه السلسلة لنسخ محتويات سلسلة رمزية الى سلسلة أخرى أي حذف القيمة القديمة للسلسلة الرمزية ووضع بدلها قيمة السلسلة الأخرى وتأخذ هذه الدالة الصيغة التالية:

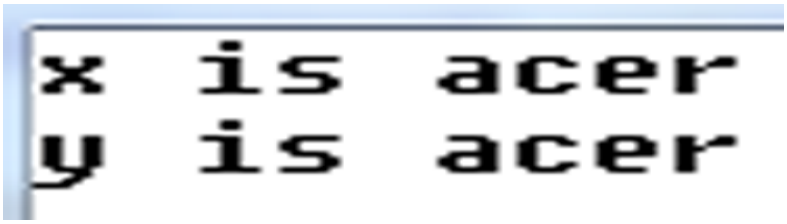
strcpy(x,y);

***برنامج لقراءة سلسلتين رمزيتين x & y ثم يقوم بنسخ عناصر y الى x**

```
#include<iostream.h>
#include<string.h>
main()
{char x[20]="toshiba",y[20]="acer";
strcpy(x,y);
cout<<"x is "<<x;
cout<<"\ny is "<<y;}
```


مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

عند تنفيذ الكود السابق تكون المخرجات كما يلي:



من الواضح أن عناصر السلسلة x قد تغيرت من "toshiba" الى"acer"

٤- الدالة ()strcmp: تستخدم هذه الدالة للمقارنة بين محتويات سلسلتين رمزيتين إذ تعيد قيمة موجبة إذا كانت السلسلة الرمزية الأولى أكبر من السلسلة الرمزية الثانية، أو تعيد القيمة(0) اذا كانت السلسلتين متساويتين، أو تعيد قيمة سالبة إذا كانت الثانية أكبرمن الأولى مع العلم أن المقارنة لا تكون من حيث عدد العناصر لكن تكون من حيث حالة الحرف الأول في السلسلة كبير أم صغير
وشكلها العام كما يلي:

```
strcmp(x,y);
```

وتقوم بعملية الطرح لأول الحروف من السلسلتين x & y حسب الجدول التالي

character	decimal	character	decimal
A	65	a	97
B	66	b	98
C	67	c	99
D	68	d	100
…	…	…	…
…	…	…	…
Z	90	z	122

أي أن الحرف الصغير(small) أكبر من الكبير ب٣٢ رقم

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

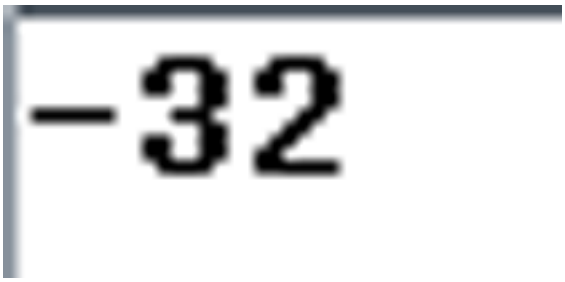
*برنامج لإدخال سلسلتين حرفيتين x & y ثم يقارن بينهما

```
#include<iostream.h>
#include<string.h>
main()
{char x[30],y[30];
cout<<"enter x : ";
cin>>x;
cout<<"enter y : ";
cin>>y;
cout<<strcmp(x,y);}
```

*برنامج لقراءة مصفوفتين x & y ثم يقارن بينهما

```
#include<iostream.h>
#include<string.h>
main()
{char x[20]="Computer",y[20]="computer";
cout<<strcmp(x,y);}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي:



لأن y أكبر من x

٥- الدالة strlwr(): تقوم هذه الدالة بتحويل عناصر سلسلة رمزية من الحروف الكبيرة الى الحروف الصغيرة وصيغتها العامة كما يلي:

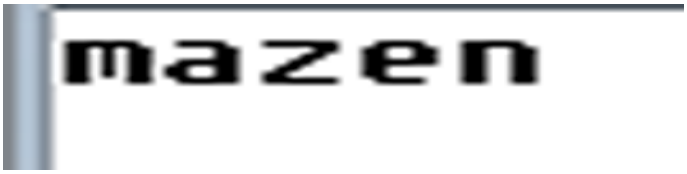
```
strlwr(x);
```

حيث x سلسلة رمزية معينة

*برنامج لقراءة سلسلة حرفية x من الحروف الكبيرة ثم تحويلها الى الحروف الصغيرة

```
#include<iostream.h>
#include<string.h>
main()
{char x[6]="MAZEN";
cout<<strlwr(x);}
```

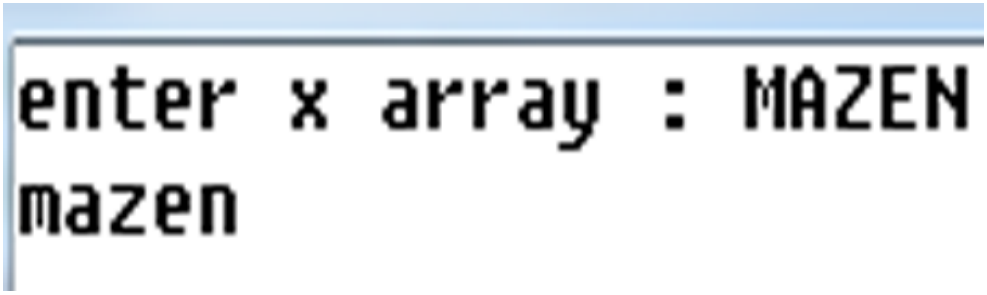
عند تنفيذ الكود السابق تكون المخرجات كما يلي:



بناء دالة تقوم بنفس العمل أي تحويل الحروف الكبيرة الى صغيرة

```
#include<iostream.h>
#include<string.h>
char lower(char x)
    {if(x>='A' && x<='Z')
    x=x+32;
    return x;}
main()
{char x[10];
cout<<"enter x array : ";
cin>>x;
for(int i=0;i<strlen(x);i++)
cout<<lower(x[i]);}
```

عند تنفيذ الكود السابق على المترجم تكون المخرجات كما يلي:



٦- الدالة strupr(): تقوم بعكس عمل الدالة strlwr() أي أنها تحول الحروف الصغيرة الى كبيرة والصيغة العامة لها كما يلي:

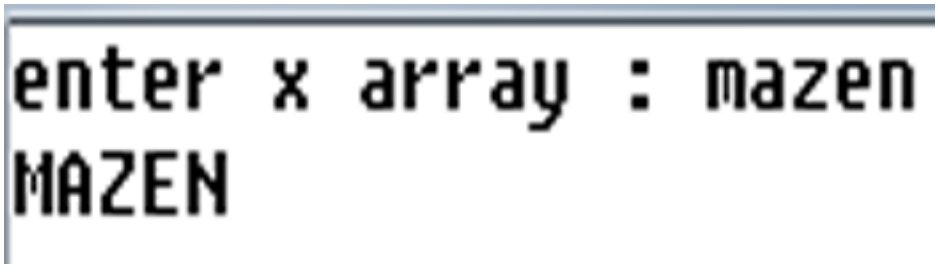
```
strupr(x);
```

حيث x سلسلة حرفية من الحروف الصغيرة

*برنامج لإدخال سلسلة حرفية من الحروف الصغيرة ثم تحويلها الى الحروف الكبيرة

```
#include<iostream.h>
#include<string.h>
main()
{char x[20];
cout<<"enter x array : ";
cin>>x;
strupr(x);
cout<<x;}
```

عند تنفيذ الكود السابق على المترجم وادخال "mazen" تكون المخرجات كما يلي:

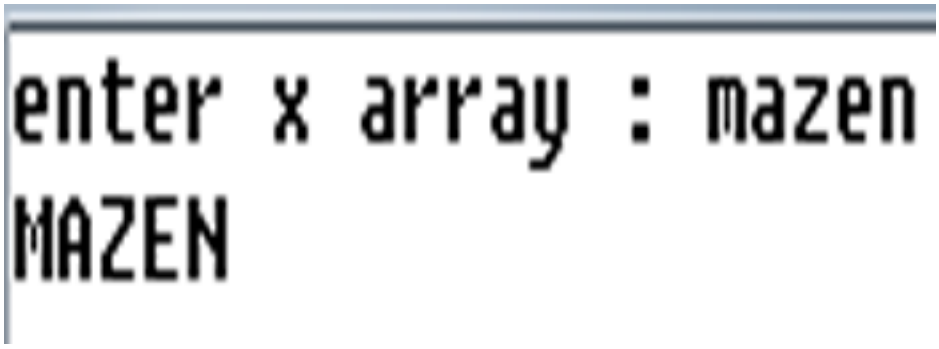


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

وإذا أردنا بناء برنامج يقوم بعمل الدالة strupr() فسيكون كما يلي:

```
#include<iostream.h>
#include<string.h>
char upper(char x)
   {if(x>='a' && x<='z')
     x=x-32;
    return x;}
main()
{char x[30];
cout<<"enter x array : ";
cin>>x;
for(int i=0;i<strlen(x);i++)
cout<<upper(x[i]);}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي:



email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

*برامج عامة خارجة عن مواضيع الكتاب:

برنامج لإنشاء مجلد جديد تحت اسم new داخل مجلد bin الخاص بالمترجم أو إنشاء ملف نصي جديد أو إيقاف التشغيل أو إعادة التشغيل أو تسجيل الخروج باستخدام الدالة system("dos instruction") التي تقع ضمن المكتبة <stdlib.h> والتي تعمل فقط على مترجم البورلاند Borland c++ وهذه الدالة تنفذ أوامر نظام التشغيل دوز ms-dos بشرط كتابتها بين علامتي تنصيص مزدوجة

```
#include<iostream.h>
#include<stdlib.h>
#include<conio.h>
main()
{int x;
cout<<"press\n1 :create new folder\n2 :new text\n3
:shutdown\n4 :reset\n5 :sign out : ";
cin>>x;
switch(x)
 {case 1 : system("md new");break;
  case 2 :system("edit");break;
  case 3 :system("shutdown -s -t 00");break;
  case 4 :system("shutdown -r -t 00");break;
  case 5 :system("shutdown -l");break;
 default:cout<<"error";break;}
getch(); }
```

md new : لإنشاء مجلد جديد باسم new والأمر edit : لتحرير نص
والأمر shutdown -s -t 00 : لإيقاف تشغيل الجهاز خلال 00 ثانية
والأمر shutdown -r -t 00 : لإعادة تشغيل الجهاز خلال 00 ثانية
والأمر shutdown -l :لتسجيل الخروج ويمكن تطبيق العديد من الأوامر

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

برنامج لإدخال كلمة السر وإخفاؤها بطباعة"" عند ادخال أي رمز

```
#include<iostream.h>
#include<conio.h>
#include<string.h>
main()
{char x[]="mazen",*y=new char;
 int i,j=0;
 cout<<"enter the pass : ";
 for(i=0;i<5;i++)
     {y[i]=getch();
      cout<<"*";}
for(i=0;i<strlen(y);i++)
     if(y[i]==x[i])
     j++;
if(j==5)
cout<<"\ntrue";
else cout<<"\nfalse";}
```

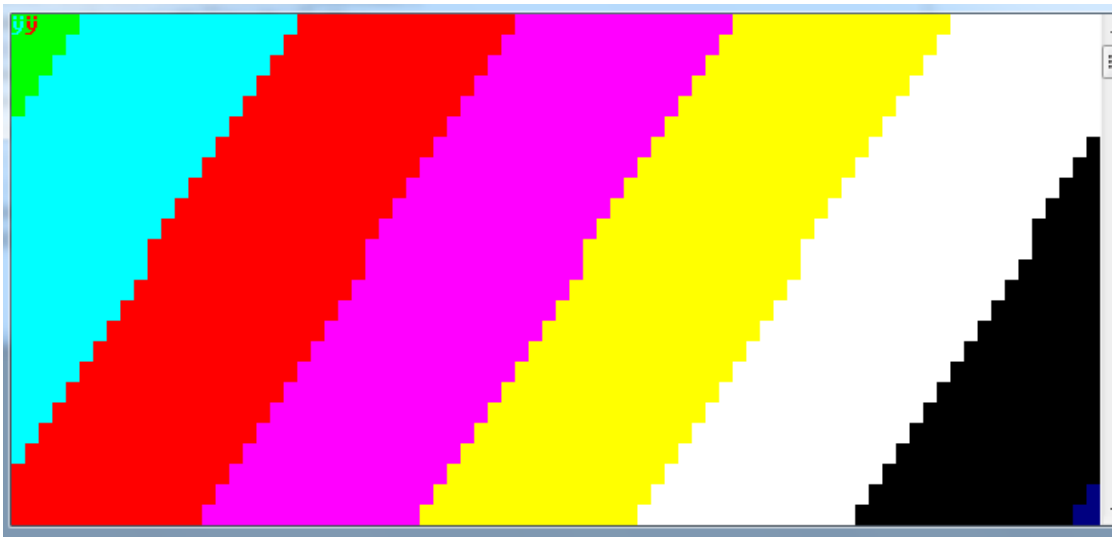
*قم بتنفيذ الكود التالي على Borland c++ ولن يعمل هذا الكود الا على البورلاند فقط

```
#include<windows.h>
main()
{ HANDLE H=GetStdHandle(STD_OUTPUT_HANDLE);
COORD A=GetLargestConsoleWindowSize(H);
COORD B={0,0};
CHAR_INFO *S=new CHAR_INFO[A.X*A.Y];
SMALL_RECT D={0,0,A.X,A.Y};
for(int i=0;i<10000;i++)
   {for(int y=0;y<A.Y;y++)
    for(int x=0;x<A.X;x++)
    S[y*A.X+x].Attributes=x+y+i;
     WriteConsoleOutputA(H,S,A,B,&D);}}
```

انظر الى السطر قبل الأخير في الكود وقم بالتلاعب بالعمليات المكتوبه بالأحمر

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

عند تنفيذ الكود السابق كما هو تكون المخرجات كما في الصورة التالية ولكن متحركة



*برنامج لتلوين خلفية الشاشة وتغييرها من اللون الأسود وكذلك تغيير لون الكتابة

```
#include<stdlib.h>
#include<iostream.h>
#include<conio.h>
main()
{
system("color 94");
cout<<"mazen_rawna";
getch();}
```

عند تنفيذ الكود السابق تكون المخرجات كما يلي :

يكون لون الخلفية أزرق لأن خانة العشرات ٩ أي ٩٠ ولون الكتابة أحمر لأن العشرات ٤ وذلك باستخدام الدالة system التي تستخدم لتنفيذ أوامر نظام dos علماً أن الأعداد التي تقبل القسمة على ١٠ من 0 الى 90 تستخدم لتغيير لون الخلفية والأعداد الأخرى بينهما تستخدم لتغيير لون الكتابة

وهناك طرق أخرى لتغيير لون الشاشة أو لون الكتابة ومنها ما يأتي

دالة تغير لون الكتابة textcolor()

تقع ضمن المكتبة conio.h وتعمل على بيئة بورلاند والشكل العام لها كما يلي:

textcolor(color number); or textcolor(color name);

أي رقم اللون أو اسمه بالحروف الانجليزية الكبيرة

والجدول التالي يوضح أكواد الألوان وأسمائها

اللون	رقم اللون	اسم اللون
أسود	0	BLACK
أزرق	1	BLUE
أخضر	2	GREEN
سماوي	3	CYAN
أحمر	4	RED
بنفسجي	5	MAGENTA
بني	6	BROWN
رمادي فاتح	7	LIGHTGRAY
رمادي غامق	8	DARKGRAY
أزرق فاتح	9	LIGHTBLUE
أخضر فاتح	10	LIGHTGREEN
سماوي فاتح	11	LIGTHCYAN
أحمر فاتح	12	LIGHTRED
بنفسجي فاتح	13	LIGHTMAGENTA
أصفر	14	YELLOW
أبيض	15	WHITE

- دالة تغير لون الخلفية textbackground()

وتقع ضمن المكتبة conio.h

وتستخدم لتغير لون خلفية الكتابة التي ستطبع بعد تحديد لون الخلفية بها وتأخذ الصيغة التالية :

textbackground(color no); or textbackground(color name);

ومعاملات هذه الدالة هي نفس معاملات الدالة السابقة ()textcolor مع ملاحظة أن الدالة ()textbackground لا تستخدم سوى الألوان من رقم 1 إلى رقم 7 المذكورين في الجدول السابق.

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

*دوال الإدخال و الإخراج التي تستخدم الألوان :
هناك مجموعة من الدوال المقابلة للدوال السابقة و التي صممت للتعامل بالألوان المحددة و كلها مسبوقة بالحرف c مثل ()cprintf و ()cputs وهي تابعة للمكتبة conio.h

```
#include<stdio.h>
#include<conio.h>
void main(){
   textbackground(BLUE);
   clrscr();
   textcolor(RED);
   cprintf("\n This text displayed with Red on Blue color");
   getch();}
```

الملحق الأول

بناءاً على اقتراحات القراء الكرام من طلاب ومعيدين قمت بإضافة الملحق الأول والهدف من هذا الملحق هو معالجة بعض الأخطاء وإضافة بعض مقترحاتكم وآرائكم وكذلك معالجة القصور وذكر الأشياء الهامة التي نسيت ذكرها في ما سبق من الدروس

٢- تابع المصفوفات الرمزية :
وقد يتم الإعلان عن المصفوفات الرمزية بعدة طرق كما يلي

```
1- char x[]="mazen";
2- char x[100]="mazen";
3- char x[10]={'m','a','z','e','n'};
```

في الطريقة رقم ١ لا داعي لتحديد حجم المصفوفة لأن المعالج يقوم بتحديده تلقائياً وكذلك لا مشكلة في ادخال المصفوفة دفعة واحدة بين علامتي تنصيص مزدوجة " "
وفي الطريقة ٢ قمت بتحديد حجم المصفوفة وأدخلتها دفعة واحدة
وفي الطريقة ٣ حددت حجم المصفوفة وأدخلت حروفها كل على حده بين علامة تنصيص مفردة وكل الطرق الثلاث صحيحة وتقوم بنفس العمل
أما اذا أردنا استخدام الطريقة ٣ دون تحديد الحجم أي إدخال المصفوفة كل حرف مستقل فلابد من إضافة الرمز '\0' كعنصر أخير لإخبار المعالج بانتهاء عناصر المصفوفة ويتم ذلك كما يلي `char x[]={'m','a','z','e','n','\0'};`

مازن الرونة للتواصل هاتف : ٧١٥٤٣٢٣٣٧ email : mazen_rawna2007@yahoo.com

واذا لم يتم اضافة الرمز '\0' الى آخر المصفوفة فان ما سيحصل هو ظهور مخرجات غريبة عند طباعة المصفوفة

أخطاء شائعة عند الإعلان عن المصفوفات الحرفية

1- char x[10];
x={'m','a','z','e','n'}; أو x="mazen";

الخطاء هنا أنه إذا أردنا ادخال المصفوفة الحرفية ادخالاً مباشراً فيجب أن يكون ذلك عند الإعلان مباشرة ولا يجوز بعد الإعلان

2- char x[]={'m','a','z','e','n'};

الخطأ هنا أنه يجب انتهاء المصفوفة الرمزية بالرمز '\0'

*اكتب مخرجات البرنامج التالي :

```
#include<iostream.h>
main()
{int x[5][5]={0},i,j;
for(i=0;i<5;i++)
{ for(j=0;j<5;j++)
 { cout<<x[i][j]<<"\t";}
  cout<<"\n";} }
```

*اكتب مخرجات البرنامج التالي :

```
#include<iostream.h>
main()
{int x[][9]={1,2,3,4,5,6,7,8,9},i;
for(i=0;i<9;i++)
cout<<x[i]<<"\n";}
```


email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

سؤال: اكتب مخرجات البرنامج التالي :

```
#include<iostream.h>
main()
{int i;
for(i=0;i<100;i++)
  {if(i%2==0)
   continue;
   cout<<i<<"\n";}
}
```

ان من ينظر الى الكود السابق يظن من الوهلة الأولى أن البرنامج يطبع الأعداد الزوجية بين الصفر والمئة لكن هذا البرنامج يطبع الأعداد الفردية فقط لأن الشرط هو اذا كان العدد زوجي continue; وهي جملة من جمل الهروب مثل break; و كذلك جملة goto واذا لم يتحقق الشرط اطبع i

- استخدام جملة goto بدل عن دوارة كما يلي:

```
#include<iostream.h>
main()
{int i=0;
mazen:
{cout<<i<<"\n";
 i++;
 if(i<100)
 goto mazen;} }
```

هذا البرنامج يطبع الأعداد من صفر الى ٩٩ بدون دوارة

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

نماذج اختبارات الترم الثاني ٢٠١٢
١- نظري (نهاية الترم- د.خالد الحسيني)

Q1

True/false :if your answer is "false", you must provide correct reason to receive full credites.

a)if we defined char *x[3]; one correct way

of copying string "Hello" to the second element

of x is strcpy(*x[1],"Hello");

b)the following statement are correct

static int z=5;

double x[z*2];

c)the following statement are correct

char x[80];

x="Hello";

Q2

Multiple choice

A .what would be printed by the following statements?

```
Int  y[3][3]={1,2,3,
              4,5,6,
              7,8,9}
cout<<*(y+2);
```

(a) 5
(b) 6
(c) 2
(d) 4 (e) Compilation/Linkage Error.

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة

B .which of the following is an incorrect string initialization?

(a) char plant[ ]="Tree";
(b) char plant [ ]={'T','R','E','E'};
(c) char plant[80]="Tree";
(d) char plant[80]={'T','R','E','E'} ;

Q3

write the output of the following program

```
#include<iostream.h>

int  f1( int x)

     {  if(x<=2)

       return 2;

      else return  2*f1(x-1);

      }

main()

{int k=4,m=6;

cout<<f1(k)<<f1(m);

}
```

Q4

a. Supposed that we have a two dimensional array int data [4][4];,please write a function void max(int *) which find and print the maximum value in array.

b. write program that prints and inputs information of 100 students ,where the information is :name , address, marks 3 subjects, and the average.

email : mazen_rawna2007@yahoo.com للتواصل هاتف : ٧١٥٤٣٢٣٣٧ مازن الرونة