



وزارة التعليم العالي
الكلية التكنولوجية بالمطرية
المعهد الفني الصناعي بالمطرية
شعبة حاسبات وشبكات

منزكرة البرمجة لغة Visual Basic6

إشراف

ح/إمام مغاورى

إثنان لا يشبعان : طالب علم وطالب مال.

المقدمة:

الحمد لله الذي علم بالقلم، علم الإنسان ما لم يعلم، أحمدده حمد الشاكرين، وأثني عليه بما هو أهله، والصلاة والسلام على معلم الناس الخير، وعلى آله وصحبه، وكل من دعا بدعوته واقتفى أثره إلى يوم الدين.

أما بعد فإن طلب العلم ونشره من أفضل القربات، التي يتقرب بها العبد من ربه، ومن أبرز الطاعات التي تعلي منزلة المسلم، وترفع قدره عند الله تعالى، ولقد أمر الله عباده بالعلم والتعلم، والتفكير والتدبر، وحذرهم من الجهل واتباع الهوى، وبين أن العلم الذي ينفع صاحبه يوم القيامة هو العلم الذي يخلص فيه العبد لمولاه، ويبتغي فيه نيل رضاه، ويتأدب فيه بأدب الإسلام، ويتخلق بأخلاق سيد الأنام، صلى الله عليه وسلم، الذي كان خلقه القرآن لذا، وجب علينا تعلم كل ما هو جديد، وفي عصرنا الحالي، لابد من مواكبة التطوير التكنولوجي لأنه أساس المستقبل، وفي الماضي، من لم يستطع القراءة والكتابة نسميه بالأمي، أما الآن فتعريف الأمي، هو من لم يستطع كتابة كود برمجى !
فى العام الماضى ,

تعلمنا سوياً بدائيات البرمجة وبدأنا فى محو أميتنا عن طريق تعلم لغة ++c هل تتذكرها ؟ ☺
بطريقة بسيطة وشيقة سنراجع سوياً على بعض الأساسيات فى الخرائط كأساس لأى عملية برمجة ولأى نوع، فى صفحتين فقط، وبعدها سنعرض منهجنا بلغة البرمجة المرئية VB6 :-
هيا بنا

لغة البرمجة المرئية Visual Basic 6

هي بيئة تطوير ولغة برمجة من مايكروسوفت تستند إلى لغة البيسك الشهيرة. وهي تصنف ضمن لغات البرمجة بالكائنات. منذ أن بدأت مايكروسوفت في إصدار الفيجوال بيسك وهي تلاقي نجاحاً باهراً وشعبية لا بأس بها بين المبرمجين نظراً لسهولة استخدامها الشديدة في مقابل التعقيد الشديد الذي يواجهه أي مبرمج يسعى لبرمجة ويندوز باستخدام السي أو السي ++، عموماً تناسب الفيجوال بيسك تطبيقات قواعد بيانات والتطبيقات المخصصة للشركات الصغيرة وبرامج الحسابات وهي مريحة وسهلة وتؤدي الغرض بالإضافة إلى أنها تسمح للمبرمج بالتركيز على حل المشكلة فغالباً ما لا يواجه صعوبات فنية أثناء كتابة برنامج بالفيجوال بيسك. ولكن بقي أن نذكر أن برامج الفيجوال بيسك لا تتم ترجمتها كاملة إلى لغة الآلة مثل السي++ أو الدلفي وإنما تترجم إلى كود وسطي يتصل مع مكتبة ربط تسمى بـ "Run Time library" واسمها MSVBM.dll مع استبدال علامات الاستفهام برقم الإصدار.

لغة البرمجة فيجوال بيسك هي لغة ذات تصميم مرئي واجهة رسومية بعكس بعض اللغات مثل (الاسمبلي) ذات الشاشة السوداء. حيث تحتوي هذه اللغة على العديد من الاوامر بداخلها ولغة سهلة التطبيق تم تطوير هذه النسخة من البرنامج عن النسخة القديمة basic والتي تعمل تحت بيئة dos إلى هذه النسخة التي تعمل تحت بيئة ويندوز. تعتمد اللغة في تطوير تطبيقاتها على الكائنات فهي تشبه العديد من لغات البرمجة الحديثة من حيث اعتمادها على الديناميكية والأحداث. تعني الديناميكية في هذه اللغة القدرة على استدعاء أي اقتران أو إجراء اعتماداً على الحدث. الحدث هو أي عملية يقوم بها مستخدم التطبيق على التطبيق مثل الضغط بزر الفارة أو الضغط على أحد أزرار لوحة المفاتيح أو حتى تحميل نموذج.

طرق حل المسائل

1. الخوارزميات (Algorithms) .

وهي مجموعة الأوامر أو الخطوات التي تكون متتالية ومنظمة لحل مسألة ما .

2. المخططات الانسيابية (Flow Charts).

وهي وسيلة بصرية مفيدة للتعبير عن الخطوات المنطقية اللازمة لحل مسألة ما ، أو يمكن القول بأنها تمثيل بياني تخطيطي لخطوات حل مسألة معينة .

3. البرامج (Programs) .

وهي تمثل التعبير المتداول بإحدى لغات البرمجة للخوارزميات أو المخططات الانسيابية . وأن احد اهم الفروق الاساسية بين الخوارزمية والبرنامج هو ان البرنامج لا بد فيه من الانضباط الدقيق بقواعد لغة البرمجة .

وتم استخدام مجموعة من الأشكال الهندسية (الرموز) في بناء المخططات الانسيابية ولكل شكل دلالة معينة ، كما يكتب داخل كل شكل العملية التي سيتم تنفيذها وأهم الأشكال المستخدمة موضحة في الجدول التالي :

الرقم	الشكل	الاستخدام	توضيح (مثال)
١		يستخدم في تمثيل بداية البرنامج ونهايته	ابدأ توقف
٢		يستخدم في تمثيل إدخال البيانات أو إخراج النتائج (طباعتها)	أدخل قيمة س اطبع قيمة س
٣		يستخدم في تمثيل معالجة البيانات مثل العمليات الحسابية	أجعل س = ص + ٢
٤		يستخدم في تمثيل اتخاذ قرار معين من خيارات (مثل المقارنات)	هل س = ٠ ؟ نعم لا
٥		تستخدم خطوط الانسياب (الاسهم) في تمثيل الانسياب المنطقي للبرنامج	هل س > ٦ ؟ نعم لا
٦		تستخدم في توصيل الاجراءات المختلفة في المخطط أو لبدء صفحة جديدة في حالة عدم الاتساع	

أهمية استخدام خرائط سير العمليات:

1. تعطي صورة كاملة لخطوات حل المسألة وتمكن المبرمج من الاحاطة بكل أجزاء المسألة.
2. تساعد المبرمج على تشخيص الأخطاء التي تقع عادة في البرنامج .
3. تيسر للمبرمج عملية متابعة المسائل التي تكتنفها الاحتمالات والتفرعات .
4. تيسر للمبرمج إدخال التعديلات على البرنامج.
5. تعتبر رسوم خرائط سير العمليات مرجعاً لحل المسائل المتشابهة .

سُبْحَانَ اللَّهِ وَالْحَمْدُ لِلَّهِ وَلِإِلَهِهِ الْإِلَهِ وَاللَّهُ أَكْبَرُ

البرمجة المرئية

المكونات

- (1) حروف A-Z ,
وهذه اللغة غير حساسة لحالة الأحرف (أى لا تفرق بين الحروف الكبيرة أو الصغيرة)
- (2) أرقام 0-9
- (3) علامات خاصة (< & = & \$ & @ & , & .)

ملاحظة هامة : إجعل الكتاب فى أى مادة هو مرجعك الأساسى , واجعل المذكرات والمراجع الخارجية للفهم والتوسع أو للتلخيص بعد المذاكرة , فالمطلوب منك فى النهاية الإستفادة أولاً وحصد درجات عالية ثانياً فلا تحيد عن هدفك 😊

فَبَيِّدُوا الْعِلْمَ بِالْكِتَابَةِ

أنواع البيانات / المتغيرات

إن البيانات التي نتعامل معها إما أرقام أو أحرف أو كلمات و الأرقام يمكن أن تكون صحيحة (أي ليس بها علامة عشرية) أو حقيقية أي بها علامة عشرية. و الحروف يمكن أن تكون حرف واحد أو أكثر من حرف و هكذا تختلف أنواع البيانات عن بعضها البعض و من الضروري معرفة أنواع البيانات و معرفة كيفية الإعلان عنها و كذلك كيفية استعمالها. ويمكن إختصار الأنواع فى السطور التالية :-

- 1 - متغير عددي
- 2 - متغير نصي
- 3 - متغير منطقي

ماهي المتغيرات

- هى عبارة عن المخازن الموجودة فى الذاكرة التى يحتجزها المبرمج ليضع فيها بعض القيم التى سيحتاجها فى تنفيذ مشروعه
- تعتبر المتغيرات هى الأساس التى يعتمد عليه المبرمج فى عمل البرنامج الذى يريد

و الجدول التالي يوضح هذه الأنواع و كذلك عدد البايث التي يشغلها كل نوع:

المتغير	نوع المتغير
integer	عدد صحيح
Long	عدد صحيح طويل
Single	عدد حقيقي ذو دقة منفردة
double	عدد حقيقي ذو دقة مزدوجة
Currency	عدد حقيقي ذو دقة منفردة
String	مجموعة من الحروف والرموز
Byte	عدد صحيح
Boolean	قيم منطقية
Date	تاريخ
Object	كائن
Variant	يصلح لجميع الأنواع "بتاع كله" ☺

Integer

يشغل 2 بايت وهى ارقام صحيحة صغيرة المدى وتتراوح بين 32767 الى -32767
long (يشغل 4 بايت وهى ارقام صحيحة كبيرة المدى وتتراوح بين 2147483648 الى -2147483648

single

يشغل 4 بايت وهى ارقام حقيقية ذات دقة بسيطة

double

يشغل 8 بايت وهى ارقام حقيقية ذات كسر عشري وذات دقة عاليه

currency

متغيرات عمله (ارقام ماليه)
تستخدم فى البرامج المالية الدقيقة وتشغل 8 بايت فى الذاكرة

string متغيرات حرفية

تستخدم لتخزين قيم البيانات الحرفية مثل الأسماء وتشمل كل حروف لوحة المفاتيح

boolean متغيرات منطقية

(true,false) او (yes,no) تستخدم لتخزين قيم البيانات المنطقية

byte

يستخدم هذا المتغير عند استخدام ارقام محددة وتتراوح بين 0,255

variant

يستخدم لتخزين اى قيمة لاي بيانات حسب نوعية البيانات نفسها بمعنى ان هذا المتغير يتشكل حسب الحاجة وحسب نوع البيانات

وهذه القيمة (default) القيم الابتدائية – الافتراضية) فى حالة عدم تحديد النوع لاي متغير.

الإعلان عن المتغيرات , حجز وتعريف المتغيرات

الإعلان عن المتغير أن تخبر “فيجول بيسك” عن اسم المتغير ونوعه ، مثل

1 Dim Name As String

الصورة العامة للأمر (نوع المتغير as اسم المتغير dim)

يجب ان يتم تعريف اى متغير قبل استخدامه فى البرنامج بامر dim وهى اختصار لكلمة dimension وتعنى ابعاد او مواصفات المتغير , الـ Name أعلاه هو اسم المتغير
اسم المتغير : هو اسم للمتغير ويجب ان يبدأ بحرف هجائى ولا يشمل حروف خاصة او مسافات
نوع المتغير : احد انواع المتغيرات السابقة

بعض من الامثلة عن حجز المتغيرات

dim n as string هنا نحجز مكان فى الذاكرة اسمه n ونوعه string حرفى
dim a as integer كما تعلمنا ان الاسم ياتى بعد كلمة dim اذن اسم المتغير هنا a ونوعه integer
dim a1 as integer هنا المتغير اسمه a1 والنوع زى السابق
dim abc as double اسم المتغير abc ونوعه double
dim Ave as single اسم المتغير Ave ونوعه single

شروط تسمية المتغيرات الثوابت (المعرفات):

- يخضع اسم المتغير لشروط معينة يجب أن تعرفها تجنباً لأخطاء قد تقع فيها و فيما يلي أوضح هذه الشروط:
- يجب أن يبدأ المتغير بحرف ثم يكمل المتغير بعد ذلك بحروف أو أرقام.
- عدم استخدام المسافات الخالية , مثل Name
- عدم استخدام أى علامة خاصة سوى الشرطة التحتية (_).
- عدم استخدام الكلمات المحجوزة فى اللغة مثل int, For .
- من الممكن أن يشتمل اسم المتغير أو المعرف حتى 40 حرف و ما زاد عن ذلك لا يلتفت إليه مترجم اللغة.
- لا تفرقة فى الفيجوال بيسك بين الحروف الصغيرة و الكبيرة فالمتغير St لا يختلف عن المتغير st فإذا استعملنا فى البرنامج يعتبر هما البرنامج متغير واحد.

اجعل من براك , بدعو لمن رباك

المؤثرات أو المعاملات Operators

المؤثرات هي الرموز التي تربط بين المتغيرات و الثوابت لإنشاء علاقة ما أو معادلة تختلف أنواع المؤثرات باختلاف وظيفة كل مؤثر. و تأخذ الأنواع الآتية:

أ) المؤثرات الحسابية Arithmetic Operators:

و هي علامات الجمع والطرح و القسمة و الضرب وتستخدم مع المتغيرات و الثوابت الرقمية. , أهم المؤثرات الحسابية وأولها التخصيص(=) فمثلا نكتب أن $z=5+4$, علامة اليساوى (=) تعنى هنا التخصيص , أى أن قيمتى $4+5$ مجموعها 9 هى تخصيص لقيمة الحرف المتواجد قبل علامة (=) أى أنها تخصه ولو قمت بعملية جمع تاليه $z=z+2$ فيكون الناتج حينها 11 باقى المعاملات موضحة فى الجدول أدناه

المعامل	المعاملات الرياضية في لغة Visual Basic	مثال
+	عملية الجمع ما يقوم به برمجياً	$3 + 4 = 7$ $z = x + y$
-	عملية الطرح	$3 - 2 = 1$ $z = x - y$
*	عملية الضرب	$3 * 4 = 12$ $z = x * y$
/	عملية القسمة مع اعطاء قيمة كسرية	$7 / 2 = 3.5$ $z = x / y$
\	عملية القسمة مع اعطاء القيمة الصحيحة فقط	$7 \setminus 2 = 3$ $z = x \setminus y$
Mod	ايجاد باقى القسمة	$7 \text{ Mod } 2 = 1$ $z = x \text{ Mod } y$
^	ايجاد الأس (القوة)	$2 ^ 3 = 8$ $z = x ^ y$

=	التخصيص : وهو جعل قيمة ما فى يمين المعادلة يتم تخصيصه لما فى اليسار (بشرط أن يكون يسار المعادلة حرف)	$z = z + i$
---	---	-------------

ب) مؤثرات المقارنة Relational Operators:

و تستخدم لمقارنة قيمتين لمعرفة هل هما متساويتين أو إحداهما أكبر أو أقل من الأخرى و هكذا. و يوضح الجدول التالي مؤثرات المقارنة و الرموز التي تستخدم بدلاً عنها.

المؤثر	الرمز	مثال	النتيجة
أكبر من	>	$10 > 8$	1
أصغر من	<	$10 < 8$	0
يساوي	=	$10 = 8$	0
لا يساوي	<>	$10 < > 8$	1
أقل من أو يساوي	<=	$10 < = 8$	0
أكبر من أو يساوي	>=	$10 > = 8$	1

اجتنب مصاحبة الكذاب فإن اضطررت إليه فلا تُصدِّقهُ

تساعدنا المعاملات المنطقية مع معاملات المقارنة بشكل كبير في التحكم بسير البرنامج حسب شروط تضعها أنت عبر جمل التحكم مثل جملة If ... Else وجمل التكرار مثل جملة For , وهذه المعاملات هي:

البوابة المنطقية	وظيفتها
A Not	ويكون الناتج واحد منطقي إذا كان المدخل A بـ 0 منطقي .
A And B	ويكون الناتج واحد منطقي إذا كان المدخلين A , B بـ 1 منطقي .
A Or B	ويكون الناتج واحد منطقي إذا كان أحد المدخلين A , B بـ 1 منطقي .

ترتيب أولويات التنفيذ للعمليات الحسابية :-

يقوم الحاسوب بتنفيذ العمليات الحسابية في تسلسل معين , طبقاً لدرجة الأولوية المحدد لكل معامل فمثلاً لو هناك معادلة كبيرة تحتوي على قسمة لجزء وجمع لجزء آخر , فإنه يبدأ بعملية القسمة أولاً يليها الجمع . وفي الحقيقة هذه ليس معلومة جديدة عنا , إنما هي ناتج للعمليات الحسابية العادية التي نستخدمها , وفي الجدول أدناه موضح الترتيب الذي يتم تنفيذ به العمليات بشكل صحيح ويليه مثال كما يجب أن يكتب في الإمتحان وبالصيغة أو الشكل الصحيح , حيث أن هذه النقطة غاية في الأهمية لما يقع فيها معظم الطلبة على الرغم من سهولتها , لنرى

المعنى	الرمز
الأقواس	()
الأس	^
العدد السالب	مثال $a = 5 + (-4)$
القسمة والضرب	/ *
قسمة الأعداد الصحيحة	\
بقاوة القسمة	Mod
الجمع والطرح	- +
معاملات المقارنه	< , > , <= , >= , =
معامل النفي (المنطقي)	Not
معامل و , (المنطقي)	And
معامل "أو" (المنطقي)	OR

ملاحظة: علامة = في أول أي (عملية حسابية) تُعنى (تخصيص) عدا ذلك فإنها تُعنى يساوي وتستخدم في عمليات المقارنه

ولا خير في حسن الجسوم وطولها إذا لم يزن طول الجسوم عقول

ترتيب العمليات (الأسبقية) كما عرضنا فى الصفحة السابقة

أثناء التنفيذ لجملة معينة تعددت فيها العمليات الحسابية يجب أن يعرف بال Interpreter بأياها يبدأ وقد تم ترتيب هذه العمليات حسب الأولوية في الجدول كما بالصفحة السابقة :

مثلاً لو طلبنا من الحاسوب أن ينفذ الجملة الحسابية التالية:

$$Z = A + B - C > D \text{ OR } (E - F^2) > g$$

حتى يستطيع فهمها يجب أن ندخلها بالشكل التالي:

$$Z = A + B - C > D \text{ OR } (E - F^2) > g$$

والآن سنتخيل ما سيقوم به المفسر (Interpreter) وقد تم تنفيذ البرنامج وقام المستخدم بإعطاء القيم التالية:

$$A = 1, B = 2, C = 3, D = 4, E = 5, F = 6, g = 7$$

وهذه الخطوة التي سيتبعها في عملية الحساب:

وهذه الخطوات التي يقوم بها تفصيلياً أثناء عملية الحساب:

(يُفضل التعويض عن القيم فى كل خطوه على حدا وعدم التعويض عن كافة القيم فى خطوه واحده)

$$Z = A + B - C > D \text{ OR } (E - \underline{36}) > g$$

$$Z = A + B - C > D \text{ OR } \underline{-31} > g$$

$$Z = \underline{3} - C > D \text{ OR } -31 > g$$

$$Z = \underline{0} > D \text{ OR } -31 > g$$

$$Z = \underline{0} \text{ OR } -31 > g$$

$$Z = 0 \text{ OR } 0$$

$$Z = 0$$

ملاحظة هامه : عدد العمليات التي تتم = عدد الخطوات (أسطر الحل) = 7 خطوات (سطور)

إجعل من يراك , يدعو لمن ربك ☺

لا تنس ذكر الله ☺

كيفية كتابة البرامج

الشكل العام لأي برنامج:-



الشكل العام لأي كود برنامج VB:-

Sub main ()	كود البداية
Dim	تعريف نوع المتغيرات
	محتوى البرنامج وأكواده التي تؤدي وظيفة معينة
End Sub	كود النهاية

دوال الإدخال و الإخراج:-

أولا : الإدخال :-

صندوق الإدخال Input message :

صيغة الإدخال في لغة الفيجوال بيسك هي InputBox , وتستخدم كما في المثال التالي :-

("الرسالة أو النص التعليمي المراد ظهوره للمستخدم") = InputBox اسم المتغير

Answer = InputBox ("من فضلك أدخل إجابتك")



في المربع الأبيض يدخل المستخدم إجابته ثم ينقر الزر OK

ويكون الشكل كالتالي

لماذا يبكي الشيخ على شبابه ولا يضحك الشاب لصباه ؟

ثانياً : الإخراج أو الطباعة :-

بفرض أن لدينا $A=5$ ونريد طباعتها على الشاشة
فيوجد أمران للطباعة على الشاشة , هما

1. أمر (MsgBox) .

ولطباعة القيمة أعلاه يكون الصيغة كالتالي

MsgBox A

الناتج سيكون 5

MsgBox "A=" & A

الناتج سيكون A= 5

حيث أن "A=" تعليمة نصية تخرج كما هي مكتوبة , وأن & هي للفصل بين المتغيرات

2. أمر (Debug.Print).

ولطباعة القيمة أعلاه يكون الصيغة كالتالي

Debug.Print A

الناتج سيكون 5

Debug.Print "A=" , A

الناتج سيكون A= 5

حيث أن "A=" تعليمة نصية تخرج كما هي مكتوبة , وأن (,) هي للفصل

وللأمر Debug.Print خيارات عديدة في تشكيل عملية الطباعة , نستعرضهم في الجدول التالي :-

بفرض أن $A=5$, $B=6$, $C=7$

شكل الطباعة	الاستخدام / المعنى	الأمر / الصيغة
5	طباعة قيمة الـ A	Debug.Print A
56	طباعة قيمة الـ A ويليها قيمة الـ B مباشرة	Debug.Print A ; B
5 6	طباعة قيمة الـ A ثم مسافة ثم قيمة الـ B	Debug.Print A , B
A = 5	طباعة قيمة الـ A يسبقها التعليمة النصية A=	Debug.Print "A ="
11	طباعة مجموعي قيمتي الـ A و الـ B	Debug.Print A+B
5 6 7	طباعة قيمة الـ A ثم مسافة ثم قيمة الـ B يليها مباشرة قيمة السطر التالي	Debug.Print A,B; Debug.Print C

اللهم أنت ربي لا إله إلا أنت خلقتني وأنا عبدك وأنا على عهدك ووعدك ما استطعت أعوذ بك من شر ما
صنعت أبوء لك بنعمتك علي وأبوء لك بذنبي فأغفرلي فإنه لا يغفر الذنوب إلا أنت

سُبْحَانَ اللَّهِ وَالْحَمْدُ لِلَّهِ وَلَا إِلَهَ إِلَّا اللَّهُ وَاللَّهُ أَكْبَرُ

الدوال القياسية والخاصة

أولاً : الدوال القياسية (المكتبية) :-

الدوال القياسية , هي دوال تم إنشائها لتقابل العمليات الرياضية مثل الجذر التربيعي والمقياس | -7 | والدوال المثلثية Sin() , Cos ()

دالة الجذر التربيعي SQR (X)	
SQR (X)	الجذر التربيعي لـ x
SQR (10 * X)	الجذر التربيعي لـ 10X
المقياس : ويعنى تجريد العدد من ساليته وجعله موجباً دائماً Num	
ABS (X)	المقياس للعدد x أو كما يُكتب رياضياً x
ABS (-5)	المقياس للعدد -5 أو كما يُكتب رياضياً -5
اللوغاريتمات Log x , Ln x	
Log10 (x)	لوغاريتم العدد x أو كما يُكتب رياضياً Log x
Log (x)	لن العدد x أو كما يُكتب رياضياً Ln x
الدوال المثلثية تُكتب كما هي Sin (x) , Cos (x)	
Sin (x)	الدالة الرياضية Sin x
Cos (x)	الدالة الرياضية Cos x
Tan (x)	الدالة الرياضية Tan x
asn (x)	الدالة الرياضية $\sin^{-1} x$
acs (x)	الدالة الرياضية $\cos^{-1} x$
Atn (x)	الدالة الرياضية $\tan^{-1} x$
Exp (x)	الدالة الرياضية E^x
دالة باقى القسمة (بالإضافة للأمثلة لها) X mod n	
14 mod 11 = 3	8 mod 9 = 8
7 mod 6 = 1	3 mod 4 = 3

سُبْحَانَ اللَّهِ وَالْحَمْدُ لِلَّهِ وَلِإِلَهِهِ الْإِلَهِ اللَّهُ أَكْبَرُ

اللهم أنت ربى, لا إله إلا أنت , خلقتنى وأنا عبدك, وأنا على عهدك ووعدك ما استطعت, أعوذ بك من شر ما صنعت, أبوء لك بنعمتك على و أبوء بذنبي فاغفر لي, فإنه لا يغفر الذنوب إلا أنت

ثانيا : الدوال الخاصة :-

الدوال الخاصة , هى دوال غير متواجد فى الرياضيات , لكن تم إنشائها من أجل العمليات البرمجية خصيصاً .

دالة Fix تحذف الجزء العشري من العدد (x)	
Fix (x) =	
Fix (7) = 7	Fix (16.41) = 16
Fix (7.9) = 7	Fix (0.9) = 0 Zero
Fix (-7.9) = -7	Fix (12.9) = 12
Fix (-7) = -7	Fix (-5.2) = -5
دالة Int تحول الى العدد الصحيح الأقل من أو يساوى الـ (x)	
Int (x) =	
Int (7) = 7	Int (-7) = -7
Int (7.9) = 7	Int (-7.9) = -8
دالة Round تُقرب العدد (x) إلى أقرب عدد n من الخانات	
Round (x , n) =	
Round (53.125 , 2) = 53.13	Round (42.144 , 2) = 42.14
Round (6.3574 , 3) = 6.357	Round (7.992 , 2) = 7.99
دالة sgn لها فى الناتج 3 احتمالات (0 , -1 , 1)	
sgn (x)	
Sgn (200) = 1	Sgn (-7.5) = -1
Sgn (-200) = -1	Sgn (0.001) = 1
Sgn (0.0) = 0	Sgn (- 0.00) = 0 ركز يا هندسه ☺

اللهم اني اسألك العافية فى الدنيا والآخرة. اللهم اني أسألك العفو والعافية فى ديني ودنياي وأهلي ومالي, اللهم استر عوراتي وآمن روعاتي, اللهم احفظني من بين يدي ومن خلفي وعن يميني وعن شمالي ومن فوقي وأعوذ بعظمتك أن أغتال من تحتي

الشروط

المعنى	الشروط
شروط العدد الفردى بإستخدام Fix	$(x/2 <> \text{Fix}(x/2))$
شروط العدد الفردى بإستخدام Int	$(x/2 <> \text{int}(x/2))$
شروط العدد الفردى بإستخدام Mod	$(x \bmod 2 <> 0)$
شروط العدد الزوجي بإستخدام Fix	$(x/2 = \text{Fix}(x/2))$
شروط العدد الزوجي بإستخدام Int	$(x/2 = \text{int}(x/2))$
شروط العدد الزوجي بإستخدام Mod	$(x \bmod 2 = 0)$
شروط العدد الذى يقبل القسمة على العدد n بإستخدام Fix	$(x/n = \text{Fix}(x/n))$
شروط العدد الذى يقبل القسمة على العدد n بإستخدام Int	$(x/n = \text{int}(x/n))$
شروط العدد الذى يقبل القسمة على العدد n بإستخدام Mod	$(x \bmod n = 0)$

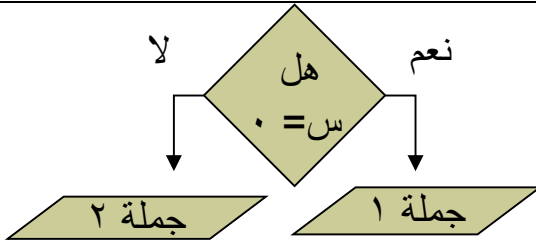
الباب الثالث

جُمْلُ التحكم أو الجُمْلُ الشرطية (التفرع)

التفرع Branching :

إذا تحقق الشرط , يتم تنفيذ الجملة 1

إذا لم يتحقق الشرط , يتم تنفيذ الجملة 2 , مع تجاهل باقى الجُمْلُ



التفرع يعني تغير مسار البرنامج. و التفرع إما أن يكون مشروط كجملة if أو غير مشروط كجملة goto .

• التفرع المشروط:

أولاً : التفرع: ifthen.....

جملة الشرط if :

تستخدم كلمة if لتنفيذ جملة أو أكثر حسب شرط معين و أبسط صورة لجملة if هي: statement الجملة المراد تنفيذها (الشرط condition) then

و معناها إذا تحقق (الشرط condition) نفذ الجملة التالية (statement) أما إذا لم يتحقق الشرط فلا تنفذ هذه الجملة و انتقل إلى التي تليها.

الجملة الشرطية - If Thenelse

تستخدم لتنفيذ أحد اختياريين و تأخذ الصورة التالية:

```
if (condition)
then
    statement1
else
    statement2
End If
```

و معناها إذا كان الشرط (condition) صحيح نفذ الجملة الأولى statment1 و إلا نفذ الجملة الثانية statment2 و هذا يعني أن تركيب if...else تستخدم لتحديد اختيار واحد من اختياريين و لا يمكن تنفيذ الاختياريين معاً كما يحدث مع جملة if وحدها.

ملاحظة : لاتنسى أن تغلق التفرع بـ End If

يتم إنهاء التفرع بجملة End If بعدد أدوات الشرط If المستخدمة , فمثلا لو استخدمت 5 شروط If لابد أن أنهى التفرع بـ 5 جملة End If وإنهاء التفرع فى هذا التفرع يكون بطريقتين كما هو موضح أدناه

إنهاء التفرع فى سطر واحد	إنهاء التفرع فى عدة أسطر
End if : End if : End If	End if
	End if
	End if

الأمثلة على ماسبق فى الصفحات بعد القادمة

ثانياً:- التفرع: Select case

إن استخدام جملة if في حالة تعدد الاختيارات لأكثر من خيارين يمثل عبئاً على المبرمج في تتبع خطوات البرنامج و يسبب بطناً نسبياً في تنفيذ البرنامج لذا استخدمنا التفرع Select..case كبديل لجملة if..then.. بطريقة أسهل كما سنرى و نستخدم بالصيغة كمال هو مبين أدناه :

واختصاراً لما سبق : التفرع Select case يستخدم في حالة تعدد الخيارات

K= inputBox "فضلاً أدخل القيمة"

```
Select case K
case 1
statement1
case 2
statement2
End case
```

معنى ذلك , انه طلب من المستخدم إدخال قيمة K , فإذا كانت هذه القيمة تساوى (1) نفذ الجملة الأولى statment1 وإذا كانت هذه القيمة تساوى (2) نفذ الجملة statment2 وهكذا

ملاحظة : يتم إنهاء هذا التفرع بجملة واحدة فقط هي End Case ولا تُكرر.

ملاحظة هامة : نستخدم Case is <50 عوضاً عن case في حالة وجود معاملات مثل < , > الفرق هنا هو إضافة كلمة is

يوجد شكل آخر لـ Case وهو

الشكل Case 0 to 49 في هذه الحالة يتم تعريف المتغير على أنه رقم فمثلاً نكتب K as string
الشكل Case 0 to 49.9 في هذه الحالة يتم تعريف المتغير على أنه رقم ورقم (عشرى) فنكتب K as Singel

مثال : أكتب برنامج يطبع تقديرات الطالب طبقاً لدرجته فإذا كان الطالب حاصل على درجات

أكبر من أو يساوى 85 اطبع "ممتاز 0" Excellent
أكبر من أو يساوى 75 اطبع "جيد جداً 0" very good
أكبر من أو يساوى 65 اطبع "جيد 0" good
أكبر من أو يساوى 50 اطبع "مقبول 0" acceptable
أقل من 50 اطبع "راسب 0" precipitate

الحل باستخدام Select case

```
Sub main ()
Dim D as Integer
R = InputBox "فضلاً أدخل درجة الطالب"
Select case R
Case is <50
MsgBox "راسب"
Case is <65
MsgBox "مقبول"
Case is <75
MsgBox "جيد"
Case is <85
MsgBox "جيد جداً"
Case is <=100
MsgBox "ممتاز"
End case
End Sub
```

الحل باستخدام If then else

```
Sub main ()
Dim D as Integer
R = InputBox "فضلاً أدخل درجة الطالب"
If D<50 then
MsgBox "راسب"
Else if D<65 then
MsgBox "مقبول"
Else if D<75 then
MsgBox "جيد"
Else if D<85 then
MsgBox "جيد جداً"
Else if D<=100 then
MsgBox "ممتاز"
End if : End if : End if : End if : End if
End Sub
```

مثال : أكتب برنامج يدخل رقمين وحرف , فإذا كان الحرف

الحل باستخدام If then else

```

Sub main ()
Dim x,y as integer
Dim m as string
x=InputBox"أدخل الرقم الأول"
y=InputBox"أدخل الرقم الثاني"
m=InputBox"فضلا أدخل علامة الحساب جمع أو ... "
If m='*' then MsgBox x*y
Else if m= '-' then
    MsgBox x-y
Else if m= '/' then
    MsgBox x/y
Else if m= '+' then
    MsgBox x+y
End If : End If : End If : End
End sub

```

* الرقم 2 فإنه يطبع يوم الأحد
* الرقم 4 فإنه يطبع يوم الثلاثاء
* الرقم 6 فإنه يطبع يوم الخميس

* الرقم 1 فإنه يطبع يوم السبت
* الرقم 3 فإنه يطبع يوم الإثنين
* الرقم 5 فإنه يطبع يوم الأربعاء
* الرقم 7 فإنه يطبع يوم الجمعة

الحل باستخدام If then else

```

Sub main ()
Dim x as integer
X=InputBox "فضلاً قّم بكتابة رقم بين 1 الى 7"
If x=1 then MsgBox "السبت"
Else if x=2 then
    MsgBox "الأحد"
Else if x=3 then
    MsgBox "الاثنين"
Else if x=4 then
    MsgBox "الثلاثاء"
Else if x=5 then
    MsgBox "الأربعاء"
Else if x=6 then
    MsgBox "الخميس"
Else if x=7 then
    MsgBox "الجمعة"
Else MsgBox "فضلاً قّم بكتابة رقم صحيح بين 1 الى 7"
End If : End If : End If : End If : End If : End If : End If :
End sub

```


الباب الرابع

الحلقات التكرارية Loops

جُمْل التكرار أو الحلقات التكرارية : هي تكرار مجموعة من الأوامر عدد معين من المرات , ويعتبر استخدام الحلقات التكرارية من أهم مميزات لغات البرمجة لأنه يفيد في إختصار حجم البرنامج من حيث عدد الجمل والأوامر وكذلك من ناحية وقت التنفيذ .

أقسام جمل التكرار:

1- جمل التكرار باستخدام For..Next

2- جمل التكرار باستخدام Do While...Loop

2- جمل التكرار باستخدام Do Until...Loop

نستعرضهم سوياً في الجدول أدناه كقاعدة وتلخيص للبواب الرابع مع شرح وظائفهم قدر الإمكان , وسنستفيد كلياً في البرامج وأعدادها قدر الإمكان من أجل الوصول الى أقل من البساطه في كتابة البرامج باستخدام الحلقات التكرارية , إضافة لأن جميع البرامج ستم كتابتها بجميع الحلقات التكرارية بشكل مُنظم وشيق , هيا بنا سنعرض خلاصة الباب في الجدول التالي والتالى فقط ☺

الحلقة Do Until Loop	الحلقة Do While Loop	الحلقة For Loop
البداية Do until (شرط التوقف)	البداية Do while (شرط الإستمرار)	الزيادة Step النهاية to البداية For
الزيادة أو مقدار الزيادة Loop I	الزيادة أو مقدار الزيادة Loop	Next I
مثال Do Until طباعة الأعداد من 1 - 100	مثال Do While طباعة الأعداد من 1 - 100	مثال For طباعة الأعداد من 1 - 100
Sub main () Dim I As Integer I = 1 Do while (I>100) I=I+1 Loop Debug.print I End Sub	Sub main () Dim I As Integer I = 1 Do while (I<=100) I=I+1 Loop Debug.print I End Sub لا يلدغ المؤمن من جحر مرتين	Sub main () Dim I As Integer For I=1 to 100 <u>step 1</u> Next I Debug.print I End Sub ملاحظة : لا يتم إضافة جملة Step في حال كان مقدار الزيادة بـ 1 , لكن إضافتها هنا للتوضيح فقط!

أمثلة متنوعة
(حاول كتابتها بـ يُمكنك ☺)

اكتب برنامج لضرب الأعداد من 1 لـ 10 وطباعة الناتج في صندوق الرسائل

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre>Sub main () Dim I , P as integer P=1 For I= 1 to 10 P=p*I Next I MsgBox P End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=1 Do While (I<=10) P=p*I I=I+1 Loop MsgBox P End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=1 Do Until (I<=10) P=p*I I=I+1 Loop MsgBox P End sub</pre>

اكتب برنامج لضرب الأعداد الفردية من 1 لـ 10

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre>Sub main () Dim I , P as integer P=1 For I= 1 to 10 step 2 P=p*I Next I MsgBox "&P*I" الناتج هو End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=1 Do While (I<=10) P=p*I I=I+2 Loop MsgBox "&P*I" الناتج هو End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=1 Do Until (I>10) P=p*I I=I+2 Loop MsgBox "&P*I" الناتج هو End sub</pre>

اكتب برنامج لضرب الأعداد الزوجية من 1 لـ 10

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre>Sub main () Dim I , P as integer P=1 For I= 2 to 10 step 2 P=p*I Next I MsgBox "&P" الناتج هو End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=2 Do While (I<=10) P=p*I I=I+2 Loop MsgBox "&P" الناتج هو End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=2 Do Until (I<=10) P=p*I I=I+2 Loop MsgBox P End sub</pre>

اكتب برنامج لضرب الأعداد التي تقبل القسمة على 4 من 1 لـ 100 مستخدماً الدالة mod

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre>Sub main () Dim I , P as integer P=1 For I= 1 to 100 If (I mod 4 =0) then P=p*I Next I MsgBox "&P" الناتج هو End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=1 Do While(I<=100) If (I mod 4 =0) then P=p*I I=I+1 Next I MsgBox P End sub</pre>	<pre>Sub main () Dim I , P as integer P=1, I=1 Do until(I>100) If (I mod 4 =0) then P=p*I I=I+1 Next I MsgBox "&P" الناتج هو End sub</pre>

إذا تم العقل نقص الكلام.

اكتب برنامج لجمع الأعداد من 1 ل 100

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre> Sub main () Dim I , Sum as integer Sum = 0 For I= 1 to 100 Sum = Sum + I Next I MsgBox "الناتج" & Sum End sub </pre>	<pre> Sub main () Dim I , Sum as integer Sum = 0 I = 1 Do While (I<=100) Sum = Sum + I I=I+1 Loop MsgBox "الناتج هو " & Sum End sub </pre>	<pre> Sub main () Dim I , Sum as integer Sum = 0 I = 1 Do Until (I >100) Sum = Sum + I I=I+1 Loop MsgBox "الناتج هو " & Sum End sub </pre>

اكتب برنامج لجمع الأعداد الفردية من 1 ل 100

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre> Sub main () Dim I , Sum as integer Sum = 0 For I= 1 to 100 step 2 Sum = Sum + I Next I MsgBox "الناتج" & Sum End sub </pre>	<pre> Sub main () Dim I , Sum as integer Sum = 0 I=1 Do While (I <=100) Sum = Sum + I I=I+2 Loop MsgBox "الناتج" & Sum End sub </pre>	<pre> Sub main () Dim I , Sum as integer Sum = 0 I=1 Do Until (I >100) Sum = Sum + I I=I+2 Loop MsgBox "الناتج" & Sum End sub </pre>

أطبع الأعداد الصحيحة المتتالية من 1 لـ 100

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre> Sub main () Dim I as integer For I= 1 to 100 Debug.print I Next I End sub </pre>	<pre> Sub main () Dim I as integer I=1 Do While (I<=100) Debug.print I I=I+1 Loop End sub </pre>	<pre> Sub main () Dim I as integer I=1 Do Until (I>100) Debug.print I I=I+1 Loop End sub </pre>

أطبع الأعداد الزوجية المتتالية من 1 لـ 100

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
<pre> Sub main () Dim I as integer For I= 1 to 100 Step 2 Debug.print I Next I End sub </pre>	<pre> Sub main () Dim I as integer I=2 Do While (I<=100) Debug.print I I=I+2 Loop End sub </pre>	<pre> Sub main () Dim I as integer I=2 Do Until (I>100) Debug.print I I=I+2 Loop End sub </pre>

إذا رأيتم العلماء على أبواب الملوك فقل بنس الملوك وبئس العلماء، وإذا رأيتم الملوك على أبواب العلماء فقل نعم الملوك ونعم العلماء.

GoTo

التركيب لـ GoTo

فقرة خارج النص (بإختصار هي مثل القفز JMP في لغة الـ Assembly تماماً "راجع معالجات يا صديقي ☺")

- تسمح بنقل تسلسل البرنامج إلى أي جملة أخرى داخل البرنامج بدون شرط وبالتالي يُطلق عليها جملة التفرع الغير شرطي .
- تستخدم في تكويني الحلقات التكرارية اللانهائية .
- حلقة تكرارية نهائية : أي أنها تكرر عدد معين من المرات .
- حلقات تكرارية لا نهائية : أي أنها تكرر عدد غير معروف (لانهائي) من المرات .

الشكل العام لجملة GOTO

GOTO Line-ID

حيث أن كلمة (Line-ID) هو معرف السطر المطلوب الانتقال إليه
فمثلا

Start :

.....

.....

Goto Start

فالبرنامج هنا يقوم بترجمة الأكواد من البداية وعندما يصل الى الجملة Goto Start فإنه يقفز
مره أخرى الى بداية البرنامج عند الكلمة Start

مثال : لطباعة الأرقام من 1 الى مالا نهائية (1,2,3,.....) أي حلقة تكرارية لانهاية باستخدام
Goto فإن البرنامج سيبدأ بالتنفيذ ولن يتوقف إلا بعد أن يقوم المبرمج بإيقافه عن طريق الضغط
على Ctrl+C أو Ctrl+Break

Sub Main ()

Dim I as integer

I=1

Start:

Debug.Print I

I = I + 1

Goto Start

End Sub

ولم أر في عيوب الناس شيئاً كنقص القادرين على التمام ☹

مثال : أكتب برنامج لحساب مضروب رقم صحيح أكبر من الصفر المستخدم في لوحة المفاتيح باستخدام أمر Goto وطباعة الناتج في صندوق الرسائل

```
Sub Main ()  
Dim K as long  
Dim I as long  
Dim J as long  
Dim F As long
```

Start:

```
K= InputBox (" أدخل قيمة أكبر من الصفر وأقل من 10 ")
```

```
If K >= 0 And K<10 Then
```

```
Goto begin
```

```
Else
```

```
MsgBox "أعد المحاولة مره أخرى , القيمة خارج النطاق"
```

```
Goto Start
```

```
End If
```

begin:

```
F = 1 : j =1
```

again:

```
If J <= K Then
```

```
F= F * J
```

```
J = J + 1
```

```
Goto again
```

```
End If
```

```
MsgBox "F هو " & K & " مضروب العدد"
```

End Sub

ملاحظة : يُفضل تعريف كل متغير في سطر كما في المثال أعلاه , وعدم تعريفهم في سطر واحد حتى لو من نفس النوع , فالتعريف التالي Dim K , I , J , F As Long غير مُفضل هنا! ☺

لا يستمتع بالجوزة إلا كاسرها ☺

الباب الخامس

المصفوفات Arrays

معنى المصفوفات

تنقسم البيانات إلى بيانات حرفية (string) وبيانات عددية صحيحة (int) وبيانات حقيقية (single)
تعريف المصفوفة : بأنها عبارة عن متغير يحمل اسما مميزا ويتكون هذا المتغير من مجموعة من العناصر المتتالية المحجوزة في الذاكرة وهي تحمل نفس نوع البيانات , للإشارة الى عنصر معين في المصفوفة نقوم بتحديد اسم المصفوفة ورقم مكان هذا العنصر في المصفوفة .
يتم تحديد عناصر المصفوفة باستخدام رقم فريد يطلق عليه فهرس Index , كل رقم من هذا الفهرس له مساحة من الذاكرة , ولذلك عليك الإعلان عن المصفوفة بالحجم المناسب حسب حاجة البرنامج حتى لا تشغل مساحة كبيرة من الذاكرة دون فائدة , يمكن الإعلان عن المصفوفة بأي نوع من البيانات حسب متطلباتك مع الأخذ في الاعتبار أن جميع عناصر المصفوفة تحمل نفس نوع البيانات .
تستخدم المصفوفة بغرض تقليل عدد الأوامر التي يمكن كتابتها داخل الكود .

أنواع المصفوفات :

هناك نوعان من المصفوفات

- **مصفوفة ذات الحجم الثابت Fixed size array**
في هذا النوع ظل حجم المصفوفة ثابت أثناء تنفيذ البرنامج Dim number (6) as integer
ف كلمة Number تشير إلى اسم المصفوفة , والرقم 6 الموجود بين الأقواس يُشير إلى أن عدد عناصر المصفوفة 6 عناصر بالترقيم من 0 إلى 5 .
 - **مصفوفة متغيرة الحجم Dynamic size array**
في هذا النوع يُمكن التعديل في حجم المصفوفة أثناء تنفيذ البرنامج (وقت التشغيل)
Dim Dynamic array () as string
ولكن قبل البدء في وضع القيم لعناصر هذه المصفوفة , عليك باستخدام الكلمة المحجوزة ReDim أولاً مع ذكر الحجم .
ReDim Dynamic array (2)
- و يمكنك Visual basic من إنشاء والتعامل مع المصفوفات سواء كانت (أحادية البعد) أو متعددة الأبعاد
- **المصفوفة أحادية البعد :**
هي عبارة عن مجموعة من القيم أو العناصر المنظمة من خلال صف واحد وعدة أعمدة أو عمود واحد وعدة صفوف
 - **المصفوفة متعددة الأبعاد :**
عبارة عن مجموعة من القيم منظمة في صفوف وأعمدة , لتحديد عنصر في مصفوفة ذات بعدين نستخدم فهرسين أحدهما يشير إلى الصف والآخر يُشير للعمود , قد تصل المصفوفة المتعددة الأبعاد في الفيجوال بيسك إلى 60 بعد , ولكن الأكثر شيوعاً في الاستخدام المصفوفة ذات بعدين والمصفوفة ذات الثلاثة أبعاد

فكرة لفهم أوسع ل معنى المصفوفات ☺

المصفوفة ذات البعد الواحد (الأحادية) مثل:

$$A = [3 \ 4 \ 5 \ 7 \ 9]$$

وتسمى مصفوفة ذات بعد واحد لأنها تتكون من صف واحد أو عمود واحد، وفيها حرف A هو اسم المصفوفة، والأرقام هي عناصر المصفوفة ويتم الإشارة إلى كل عنصر برقم العنصر أي بترتيبه داخل المصفوفة على أن يبدأ العد بالرقم صفر كما يلي:
العنصر $A[0]$ يساوي 3 والعنصر $A[1]$ يساوي 4 والعنصر $A[2]$ يساوي 5.

والمصفوفة ذات البعدين (الثانية) تأخذ الشكل التالي:

[5 4 2]

[5 9 3]

[2 1 0]

وتسمى المصفوفة 3×3 أي 3 صفوف و3 أعمدة ويتم الإشارة إلى عناصر المصفوفة برقم الصف ورقم العمود الذي يقع عندهما العنصر كما يلي:

العنصر $C[0][0]$ يساوي 5

العنصر $C[0][1]$ يساوي 4

العنصر $C[0][2]$ يساوي 2

العنصر $C[2][1]$ يساوي 9

والخلاصة أن المصفوفة هي مجموعة من العناصر سواء ذات بعد واحد أو بعدين بشرط أن تكون جميع العناصر من نوع واحد وفيما يلي سنوضح كيفية الإعلان عن المصفوفة وكيفية التعامل مع عناصرها.

أَحْسِنُ إِلَى النَّاسِ تَسْتَعِيدُ قُلُوبَهُمْ

أمثلة على المصفوفة أحادية البعد

أكتب برنامج VB يدخل درجات فصل دراسي به 100 طالب في احد المواد في مصفوفة أحادية Score, أحسب مجموع درجات الطلاب الناجحين فقط (أي الحاصلين على أكبر < من المتوسط)

خطوات الحل المُتبعة (مجموع الدرجات , ثم المتوسط , ثم جمع درجات الناجحين)

```
Sub Main ( )
Dim Sum , I , Score , sum1 (100) as integer
Dim Ave as single
Sum = 0
For I= 0 to 99
Score ( I ) = InputBox "فضلا أدخل درجات الطلاب "
Sum = Sum + Score (I)
Next I

Ave = Sum / 100

Sum1=0
For I =0 to 99
If ( Score (I) > Ave ) then Sum1=Sum1+ Score (I)
Next I

MsgBox "الناتج هو" & Sum1
End sub
```

أكتب برنامج VB يحسب الجذر التربيعي لمجموع مربعات العناصر ذات التسلسل الزوجي فقط من مصفوفة أجدادية البعد رقمية مكونة من 100 رقم

```
Sub Main ( )
Dim Sum , I as integer
Q as single
Sum = 0
For I= 0 to 99 step 2
Score ( I ) = InputBox "فضلا أدخل القيم "
Sum = Sum + Score (I)^2
Next I

Q = SQR (Sum)

MsgBox "الناتج هو" & Q
End sub
```


أنشئ (أدخل أو خزن) مصفوفة ثنائية رقمية K تحتوى على 2 صف و 3 عمود, أوجد
أ - أجمع عناصر هذه المصفوفة وإطبع الناتج ؟
ب - أوجد حاصل ضرب عناصر المصفوفة وإطبع الناتج ؟
ج - طباعة عناصر المصفوفة ؟

= < ملاحظة : المصفوفة الثنائية عبارته عن 2 حلقة تكرارية متداخلة إحداها للصف وأخرى للعمود
يتم تقسم المسألة إلى أجزاء أثناء الحل كما هو مكتوب أدناه

```
Sub Main ( )  
Dim I , J , Sum , P as integer  
Dim K(2,3) As integer
```

بداية من هنا (إنشاء عناصر المصفوفة) ---- <

```
For I = 0 To 1  
    For J = 0 To 2  
        K (I,J) = InputBox "فضلا أدخل عناصر المصفوفة"  
    Next J  
Next I
```

بداية من هنا (جمع عناصر المصفوفة) ---- <

```
Sum= 0  
For I = 0 To 1  
    For J = 0 To 2  
        Sum = Sum + K (I,J)  
    Next J  
Next I  
MsgBox "ناتج جمع العناصر هو" & Sum
```

بداية من هنا (حاصل ضرب عناصر المصفوفة) ---- <

```
P = 1  
For I = 0 To 1  
    For J = 0 To 2  
        P = P * K (I,J)  
    Next J : Next I ☺  
MsgBox "ناتج حاصل ضرب عناصر المصفوفة هو" & P
```

بداية من هنا (طباعة عناصر المصفوفة) ---- <

```
For I = 0 To 1  
    For J = 0 To 2  
        Debug.print K (I,J) & " " ;  
    Next J  
    Debug.print vbNewLine  
Next I  
End sub
```

أمثلة متنوعة (أخرى ☺)

أكتب برنامج يطبع الأعداد التي تقبل القسمة على 7 للأعداد في المدى 50 الى 150 باستخدام Do while بطريقتين مختلفتين!

Do While Loop باستخدام	ثانيا : باستخدام Do While Loop	أولاً : حل إضافي باستخدام For loop
<pre> Sub main () Dim I as integer I = 50 Do While (I<=150) If (I mod 7 = 0) then Debug.print I End if I = I + 1 Loop End sub </pre>	<pre> Sub main () Dim I as integer I = 56 Do While (I<=147) Debug.print I I = I + 7 Loop End sub </pre>	<pre> Sub main () Dim I as integer For I= 56 to 147 step 7 Debug.print I Next I End sub </pre>

أمثلة متنوعة ☺

<pre> Sub main () Dim I , sum as integer I = 1 : Sum = 0 Begin : If (I <= 50) then Sum = sum + I I = I + 1 Go to Begin End if MsgBox "الناتج هو" & Sum End sub </pre>	<h3 style="margin: 0;">أكتب برنامج لحساب الأعداد من 1 الى 50 بإستخدام Goto وطباعة الناتج في صندوق الرسائل <====</h3>
--	---

أكتب برنامج لحساب مضروب رقم صحيح أكبر من الصفر يدخله المستخدم من لوحة المفاتيح , ويطبع الناتج في صندوق الرسائل
بإستخدام أمر Go to ومره أخرى بإستخدام الحلقة التكرارية For

Do Until Loop باستخدام	ثانيا : باستخدام Do Until Loop	أولاً : باستخدام For loop
<pre> Sub main () Dim K , F , I As integer Start : K = InputBox ("فضلا أدخل رقم أكبر من الصفر وأقل من عشرة ") If k > 0 and K < 10 then Goto Begin Else MsgBox"فضلا أعد كتابة رقم صحيح" Goto Start End if Begin : F = 1 I = 1 again : If j <= k then F = F * I I = I + 1 Goto again End if MsgBox "F هو" & F & "مضروب العدد" & k End sub </pre>	<pre> Sub main () Dim K , F , I As integer Start : K = InputBox ("فضلا أدخل رقم أكبر من الصفر وأقل من عشرة ") If k > 0 and K < 10 then Goto Begin Else MsgBox"فضلا أعد كتابة رقم صحيح" Goto Start End if Begin : F = 1 For I = 1 To K F = F * I Next I MsgBox "F هو" & F & "مضروب العدد" & k End sub </pre>	<pre> Sub main () Dim K , F , I As integer Start : K = InputBox ("فضلا أدخل رقم أكبر من الصفر وأقل من عشرة ") If k > 0 and K < 10 then Goto Begin Else MsgBox"فضلا أعد كتابة رقم صحيح" Goto Start End if Begin : F = 1 For I = 1 To K F = F * I Next I MsgBox "F هو" & F & "مضروب العدد" & k End sub </pre>

اللهم أنت ربى لا اله إلا أنت خلقتنى وأنا عبدك وأنا على عهدك ووعدك ما استطعت , أعوذ بك من شر ما صنعت , أبوء لك بنعمتك على وأبوء بذنبي فأغفرلى فإنه يغفر الذنوب إلا أنت

أكتب برنامج فيجوال بيسك باستخدام إحدى الحلقات التكرارية لطباعة الأتى

$P = 1/3 * 1/6 * 1/9 \dots\dots\dots 1/99$	$Sum = \underline{1/2} + 1/8 + 1/16 + \dots 1/128$	$Sum = 1/2 + 1/4 + 1/6 + \dots 1/200$
Sub main () Dim I , P as integer P = 1 For I= 3 to 99 step 3 P = P * 1 / I Next I MsgBox "الناتج هو" & Sum End sub	Sub main () Dim I , sum as integer <u>Sum = 1/2</u> For <u>I= 8</u> to 128 step 8 Sum = Sum + 1 / I Next I MsgBox "الناتج هو" & Sum End sub	Sub main () Dim I , sum as integer Sum = 0 For I= 2 to 200 step 2 Sum = Sum + 1 / I Next I MsgBox "الناتج هو" & Sum End sub

أكتب برنامج لجمع الأعداد التى تقبل القسمة على 2 و 3 معاً فى الأعداد من 1 الى 100 باستخدام 3 حلقات تكرارية ☺

ثانياً : باستخدام Do Until Loop	ثانياً : باستخدام Do While Loop	أولاً : باستخدام For loop
Sub main () Dim I , Sum as integer Sum = 0 I = 1 Do Until (I>100) If (I mod 2 = 0 and I mod 3 = 0) Then Sum = Sum + I End if I=I+1 Loop MsgBox "الناتج هو" & Sum End sub	Sub main () Dim I , Sum as integer Sum = 0 I = 1 Do While (I<=100) If (I mod 2 = 0 and I mod 3 = 0) Then Sum = Sum + I End if I=I+1 Loop MsgBox "الناتج هو" & Sum End sub	Sub main () Dim I , Sum as integer Sum = 0 For I= 1 to 100 If (I mod 2 = 0 and I mod 3 = 0) Then Sum = Sum + I End if Next I MsgBox "الناتج هو" & Sum End sub

صاغ مختلفاً لنفس السؤال (لنفس البرنامج)

* أكتب برنامج فيجوال بيسك لطباعة جدول الضرب للأعداد من 1 إلى 5 مضروبة فى الأعداد من 1 إلى 5 باستخدام حلقة تكرارية ؟
* أكتب برنامج فيجوال بيسك يحسب حاصل ضرب الأعداد من 1 إلى 5 مضروبة فى الأعداد من 1 إلى 5 ؟

لا إله إلا الله	الحل : باستخدام الحلقة For	مُحمد رسول الله
Sub main () Dim I as integer Dim j as integer For I = 1 to 9 For J = 1 to 9 Debug.Print I ; "*" ; J ; "=" ; I*J Next J Next I End sub		

أطبع الأرقام من (10) إلى (1) تنازلياً

أولاً : باستخدام For loop	ثانياً : باستخدام Do While Loop	ثانياً : باستخدام Do Until Loop
Sub main () Dim I as integer For I= 10 to 1 step -1 Debug.Print I Next I End sub	Sub main () Dim I as integer I = 10 Do While (I >= 1) Debug.Print I I = I - 1 Loop End sub	Sub main () Dim I as integer I = 10 Do While (I < 1) Debug.Print I I = I - 1 Loop End sub

أكتب برنامج لإدخال رقم من (0) إلى (10000) , ثم طباعة عدد خانات الرقم المُدخل

سُبْحان الله العظيم	الحل باستخدام If ...ThenElse	سُبْحان الله وبحمد
Sub main () Dim A , B as integer Start : A = InputBox ("أدخل عدد أكبر من أو يساوى الصفر واقل من أو يساوى الـ 10000") If A >= 0 and A <= 10000 Then Goto Begin Else MsgBox "الرقم الذى تم إدخاله خارج النطاق المحدد , فضلا أعد المحاولة مره أخرى" Goto Start End If If A < 10 Then B = 1 Else If A < 100 Then B = 2 Else If A < 1000 Then B = 3 Else If A < 10000 Then B = 4 Else B = 5 End If End If End If End If MsgBox "خانة" & B & "الرقم المُدخل له" End Sub <p>ملاحظة : بداية من هذا السطر , هنا للحماية فقط ويمكنك عدم كتابتها , لكن لاتنسى عملية الإدخال إذا أردت عدم كتابة هذه الأسطر</p> <p>ملاحظة 2 : إلى هنا , نهاية الحماية من إدخال رقم خاطئ , والأسطر التالية هو البرنامج الفعلى ☺</p> <p>ملاحظة : يُمكن إنهاء الجملة الشرطية IF فى سطر واحد فقط هكذا : End if : End if : End if : End if , لكن فى عدة أسطر سيكون أفضل يا مهندسوا المستقبل ! ☺</p>		

إذا لم يكن عون من الله للفتى , فأول ما يُجنى عليه إجهاده

سُبْحان الله وبحمده عدد خلقه وزنة عرشه ومداد كلماته

إبتسم ☺

النظري

س1: ما المقصود بلغة البرمجة وأشرح أنواعها ؟

س2: يمر البرنامج المكتوب باللغات عالية المستوى بثلاث مراحل , إشرحها ؟

س3: أشرح ما تعرفه عن تقنيات البرمجة ؟

س4: ما المقصود بلغة الفيجوال بيسك وماهي مزاياها ؟

س5: ما المقصود بالبرمجة المسيرة بالأحداث ؟

س6: ما هي خطوات كتابة البرنامج ؟

س7: ما هي (الوحدات النمطية - الوظائف - الخصائص - الأحداث - الإجراءات العامة) ؟

س8: ما هي الأوضاع التي يمر بها البرنامج ؟

س9: أشرح المقصود بكل من (المعرفات - المتغيرات - الثوابت) وماهي شروط تسمية المتعرفات ؟

س10: ما هي أنواع البيانات ؟

س11: ما هي كيفية الإعلان عن المتغيرات والثوابت ؟

س12: تكلم عن نوعان من الحلقات التكرارية ؟

س13 : ما المقصود بالمصفوفة وما هي أنواعها ؟

تعريف المصفوفة : المصفوفة عبارة عن متغير يحمل اسما مميزا ويتكون هذا المتغير من مجموعة من العناصر المتتالية المحجوزة في الذاكرة وهي تحمل نفس النوع (من البيانات), للإشارة الى عنصر معين في المصفوفة نقوم بتحديد اسم المصفوفة ورقم مكان هذا العنصر في المصفوفة .
يتم تحديد عناصر المصفوفة باستخدام رقم فريد يطلق عليه فهرس Index , كل رقم من هذا الفهرس له مساحة من الذاكرة , ولذلك عليك الإعلان عن المصفوفة بالحجم المناسب حسب حاجة البرنامج حتى لا تشغل مساحة كبيرة من الذاكرة دون فائدة , يمكن الإعلان عن المصفوفة بأى نوع من البيانات حسب متطلباتك مع الأخذ في الاعتبار أن جميع عناصر المصفوفة تحمل نفس نوع البيانات .
تستخدم المصفوفة بغرض تقليل عدد الأوامر التي يمكن كتابتها داخل الكود .

أنواع المصفوفات :


هناك نوعان من المصفوفات

- **مصفوفة ذات الحجم الثابت Fixed size array**
في هذا النوع ظل حجم المصفوفة ثابت أثناء تنفيذ البرنامج Dim number (6) as integer
ف كلمة Number تشير إلى اسم المصفوفة , والرقم 6 الموجود بين الأقواس يُشير إلى أن عدد عناصر المصفوفة 6 عناصر بالترقيم من 0 الى 5 .
- **مصفوفة متغيرة الحجم Dynamic size array**
في هذا النوع يُمكن التعديل في حجم المصفوفة أثناء تنفيذ البرنامج (وقت التشغيل)
Dim Dynamic array () as string
ولكن قبل البدء في وضع القيم لعناصر هذه المصفوفة , عليك باستخدام الكلمة المحجوزة ReDim أولاً مع ذكر الحجم .
ReDim Dynamic array (2)

و يمكنك Visual basic من إنشاء والتعامل مع المصفوفات سواء كانت (أحادية البُعد) أو متعددة الأبعاد

- **المصفوفة أحادية البُعد :**
هي عبارة عن مجموعة من القيم أو العناصر المنظمة من خلال صف واحد وعدة أعمدة أو عمود واحد وعدة صفوف .
- **المصفوفة متعددة الأبعاد :**
عبارة عن مجموعة من القيم منظمة في صفوف وأعمدة , لتحديد عنصر في مصفوفة ذات بعدين نستخدم فهرسين أحدهما يشير الى الصف والآخر يُشير للعمود , قد تصل المصفوفة المتعددة الأبعاد في الفيچوال بيسك الى 60 بعد , ولكن الأكثر شيوعاً في الاستخدام المصفوفة ذات بعدين والمصفوفة ذات الثلاثة أبعاد

نصائح وحيل ☺ :-

- 1 - إذا طُلب في البرنامج استخدام دالة خاصة (Mod , Int , Fix) فإن الحلقة التكرارية تبدأ ببداية المدى وتنتهي بنهاية المدى وحجم الزيادة يكون (1) صحيح
- 2 - بعد بداية البرنامج بـ Sub main () , إترك من سطرين لـ 3 سطور فارغين ستكون بحاجة اليهم لتعريف المتغيرات أو أى متغيرات أو معادلات تريد إضافتها .
- 2- فى الحلقة التكرارية For , لا يتم إضافة جملة Step = 1 فى حالة كانت الحلقة التكرارية تزيد بمقدار واحد صحيح .
- 3 - فى الحلقة التكرارية For إنتبه دائما ولا تنسى أنها تنتهى بـ Next I
- 4 - فى الحلقة التكرارية For , يمكنك إنهاؤها بـ Next I أو Next فقط , والفرق بين الإثنين هو إختلاف نوع إصدار البرنامج الذى تعمل عليه , لذا يفضل دائما عند كتابتها فى الإمتحان أن تكون Next I (ريح بالك ☺) .
- 5 - فى الحلقة التكرارية Do While و Do Until , لا تنسى أنها تنتهى بـ Loop
- 6 - إذا طُلب منك تنفيذ برنامج ب الحلقة التكرارية Do While و Do Until , قم بتنفيذ البرنامج بأحد الحلقات وقم بتكراره فى الحلقة التالية مع تغير اسم الحلقة والشرط , مع الإبقاء على نفس صيغة البرنامج بدون تعديل ! (وفر وقتك يا هندسه )
- 6 - لا تنسى تعريف المتغيرات عن طريق السطر الثانى فى البرنامج Dim X As Integer/String وقم بتعريف كل متغير فى سطر واحد مستقل , هذا هو المفضل هنا ! ☺
- 7- لا تنسى إنهاء البرنامج الخاص بك بجملة End Sub
- 8- إجعل كتابك مرجعك الأساسي .
- 9 - تذكر أنه (عند الإمتحان يكرم المرء أو يُهان)
- 10 - إستعينوا بالصبر والصلاه ☺

وأخيراً لا تنسى الشروط التالية:-

الشرط	الصيغة
شرط العدد الفردى	$I \text{ Mod } 2 <> 0$
شرط العدد الزوجي	$I \text{ Mod } 2 = 0$
شرط العدد الذى يقبل القسمة على N	$I \text{ Mod } N = 0$

لا اله الا الله - محمد رسول الله

إثنان لا يشبعان : طالب علم وطالب مال.

لصق الأكواد الجاهزة من الانترنت داخل مشروعك هو مثل مضغ علكة وجذتها مرمية في الشارع" - مايك جونسون

"Pasting code from the Enternet into production code is like chewing gum found in the street." - Mike Johnson

بحمد الباري ونعمة منه وفضل ورحمه نضع قطراتنا الاخيره فى هذا العمل المتواضع -
بين تفكر وتعقل في البرمجة وقد كانت رحلة جاهده للارتقاء بدرجات العقل ومعرّاج الافكار
فما هذا الاجهد مقل ولا أدعي فيه الكمال ولكن عذرى انى بذلت فيه قصارى جهدى فان اصبحت
فذاك مرادى وان أخطئت فلى شرف المحاولة والتعلم ولا أزيد على ما قال عماد الاصفهاني:
(رايت انه لا يكتب انسان كتابا في يومه إلا قال في غده لو غير هذا لكان احسن ولو زيد كذا لكان يستحسن ولو قدم هذا
لكان افضل ولو ترك هذا لكان اجمل وهذا من اعظم العبر وهو دليل على استيلاء النقص على جملة البشر..)
وأخيراً بعد أن تقعدنا باليسير في هذا المجال الواسع
أملين أن ينال القبول ويلقى الاستحسان وفقني الله وإياكم لما فيه صالحنا جميعاً.
وصل اللهم وسلم على سيدنا وحبيبنا محمد وعلى آله وصحبه وسلم.

أسألكم الدعاء لى فى ظهر الغيب