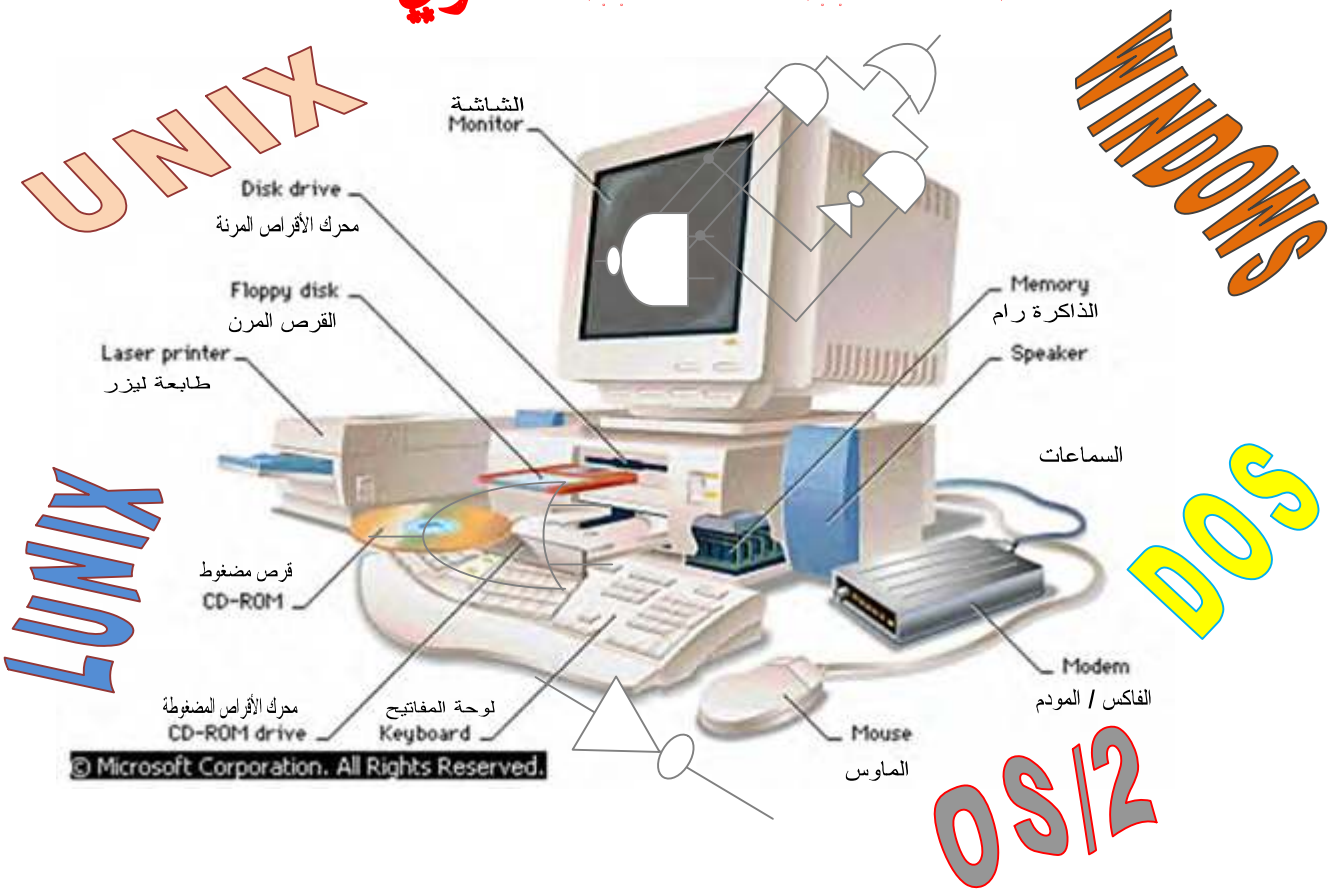
عِلموا النشىء علماً يستبين به سُبل الحياة وقبل العِلم أخلاقا

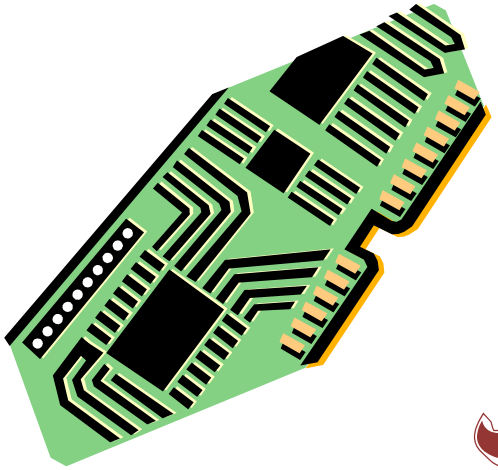
علوم الحاسوب

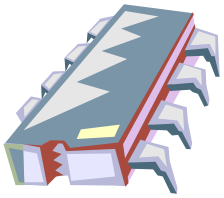
الصف الثالث الثانوي



الباب الثاني

2





بنائيات

البيانات

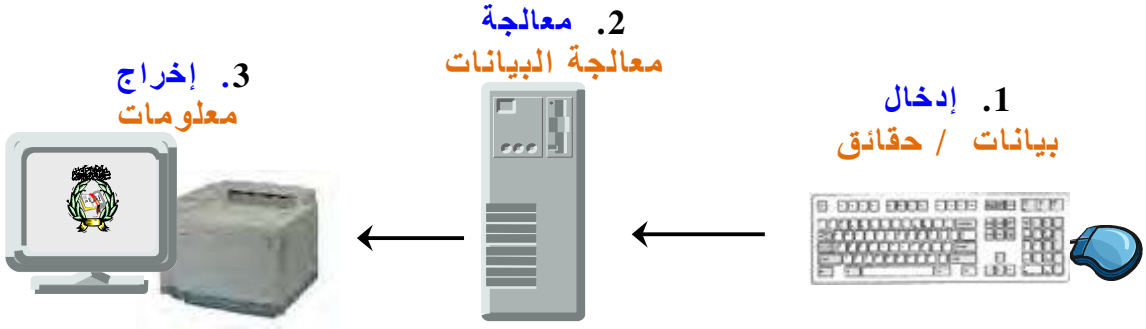
- **جهاز الكمبيوتر**

جهاز إلكتروني يقوم **بتخزين** ( ) البيانات، ثم القيام **باسترجاعها** (إخراجها) ، **المُعالجات** المطلوبة عليها وهي في **صورة رقمية ثنائية (1/0)**

.

- **دورة معالجة البيانات**

هناك ثلاثة عمليات ضرورية يقوم بها الكمبيوتر لمعالجة البيانات.



1. **إدخال البيانات**

يتم فيها إدخال أو كتابة البيانات داخل الكمبيوتر (**البيانات** : عبارة عن حقائق خامة لم تتم فيها عملية المعالجة).

2. **معالجة البيانات المُحَوسَب**

هي عملية تخزين البيانات ثم القيام باسترجاعها بعد إجراء كافة المعالجات المطلوبة عليها. وعملية استرجاع أو معالجة البيانات يمكن أن تتم بعدة طرق منها :-

1. **مُرتبة** تصاعدياً أو تنازلياً، بالحروف أو الأرقام أو الأسماء أو الأعمار.. الخ.
2. **مُصَنفة** حسب الجنس أو المهنة أو البلد أو الدرجة العلمية .. الخ.
3. **مُحَوسبة** أجريت عليها بعض العمليات الرياضية أو المنطقية أو الإحصائية.
4. **مُرسَلة** يتم إرسالها إلى جهات أخرى عن طريق الفاكس أو الإنترنت.
5. **مَرسُومة** تحول إلى رسومات أو جداول بيانية.
6. **مُشفرة** تُشفر لغرض السرية والخصوصية.

3. **إخراج المعلومات**

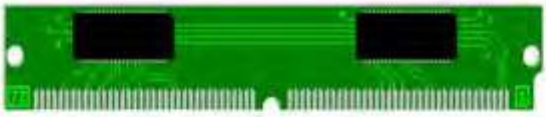
هي المعرفة الناتجة من معالجة البيانات والتي تُمكن

.

- **الخوارزميات / بنائيات البيانات**

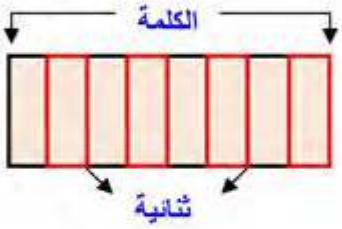
هي الطريقة التي تُمكن من تخزين البيانات ومعالجتها بكفاءة عالية **الكفاءة** : .

1.
2. أقل مساحة تخزينية ممكنة.



- **ذاكرة الكمبيوتر**

تتكون ذاكرة الحاسوب من عدد كبير من مواقع تخزين البيانات ، وهذه المواقع متساوية ، وكل موقع يتكون من عدد ثابت من الثنائيات أو البتات (Bits)، وكل موقع يسمى بالكلمة.



- **الكلمة Word**

هي موقع تخزن فيه البيانات داخل الذاكرة ، وهذا الموقع يحتوي بداخله على عدد ثابت من الثنائيات ، ويختلف هذا العدد من حاسوب لآخر حسب قوة وحجم الحاسوب، فهناك مثلاً 8 و 16 و 32 و 64 و 128 ثنائية للكلمة..

- **البايت Byte**

وحدة تقاس بها السعة التخزينية للذاكرة والأقراص ، ويرمز له بالحرف (B) .
(1 بايت = 8 ثنائيات - 8 ثنائيات أو بتات يمكن أن تخزن فيها حرف أو رمز أو رقم)

- **الثنائية / البت Bit**

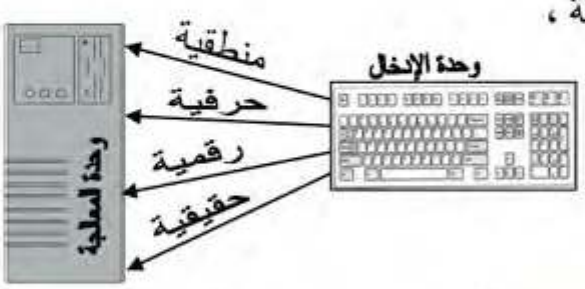
هي أصغر وحدة تخزينية في ذاكرة الحاسوب ،(8 ثنائيات = 1 بايت).

- **العنوان في الذاكرة**

العنوان هو مكان داخل الذاكرة تخزن فيه الكلمة، وعن طريق هذا العنوان يمكن الوصول إلى مكان تلك الكلمة. أما في حالة الكلمات الطويلة الثنائيات فهناك عنوان داخل عنوان الكلمة يوصل إلى الرمز أو البايت المطلوب داخل تلك الكلمة.

أنواع البيانات في لغات البرمجة

تتفق جميع لغات البرمجة على تقسيم البيانات إلى أربعة أنواع هي :-



1. **الأعداد الرقمية Integer**

هي الأعداد الصحيحة السالبة أو الموجبة ،
(**مثال** : 1 ، 90 ، -5 ، 67).

2. **الأعداد الحقيقة Real**

هي الأعداد التي بها خانة عشرية ،
(**مثال** : 0.2 ، 2.35 ، -1.6).

3. **البيانات الحرفية Character**

كل حروف الطباعة والعلامات والأرقام ، (**مثال** : أ ي ، - : ؟ , < 8ص @).

4. **البيانات المنطقية Logical / Boolean**

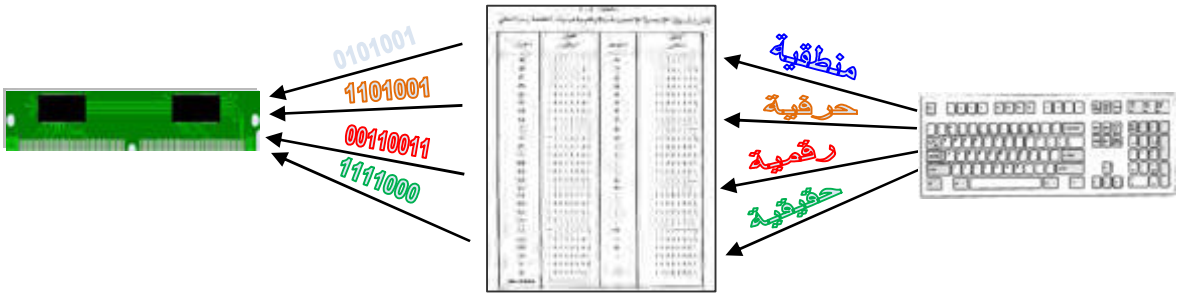
كل البيانات التي تأخذ قيمتي خطأ أو صواب، (**مثال** : 0 أو 1).

- **الترميز**

هي عملية تحويل أنواع البيانات المختلفة إلى صورة ثنائية.

- **جدول الترميز**

عبارة عن جدول، يقوم بترميز الأعداد الرقمية والحقيقية والبيانات الحرفية والمنطقية وذلك بتحويلها إلى صورة ثنائية حتى تتمكن كل موقع من مواقع الذاكرة إن كان به بيانات ذاكرة الكمبيوتر من التعامل معها.



ترميز أنواع البيانات المختلفة في لغات البرمجة

مقدمة

هي الطريقة التي يتعامل بها الحاسوب مع البيانات المختلفة عند تحويلها صورة ثنائية وتخزينها في الذاكرة تختلف طريقة تمثيل أنواع البيانات من نوع على حسب نوع البيانات المدخلة :

أولاً – تمثيل الأعداد الرقمية

هنالك ثلاث خوارزميات مشهورة تستخدم في ترميز الأعداد الرقمية هي :

1. خوارزمية علامة القيمة.
2. خوارزمية الإكمال / الإتمام الأحادي.
3. خوارزمية الإكمال / الإتمام الثنائي.

☜ :

1. أن الثنائية الأولى في أقصى يسار الكلمة (أقصى اليمين في اللغة العربية) تكون لتحديد علامة الرقم السالب و الموجب.
2. يُمثِّل الرقم صفر "0" علامة الموجب ، والرقم واحد "1" علامة السالب.
3. يتم جمع الأرقام الثنائية الموجبة بنفس طريقة جمع الأرقام الثنائية.

☜ :

1. جمع الأرقام الثنائية السالبة -(موضحة في 1) :

هي علامة العدد الثنائي السالب (+ -).

1. تمثيل قيم الأرقام الموجبة

+ 2 =	10	=	
+ 7 =	111	=	
+ 9 =	1001	=	
+ 0 =	0	=	

الرقم
أقصى يسار
الكلمة يمثل

(السالب
الموجب)
ويعرف

بثنائية

العلامة.

2. تمثيل قيم الأرقام السالبة

- 2 =	10	=	
- 7 =	111	=	
- 9 =	1001	=	
- 0 =	0	=	

1. علامة القيمة

1. قيم الأرقام السالبة مثل قيم الأرقام الموجبة.
2. عيب هذه الخوارزمية أظهرت بأن " +0 – 0" هما قيمتين مختلفتين في حين أنهما قيمة واحدة.

2. الإكمال الأحادي

1. تحول أي ثنائية قيمتها صفر إلى واحد وأي ثنائية قيمتها واحد إلى ( ).
2. لها نفس عيب الطريقة الأولى في .
3. ولها ميزة على الطريقة الأولى وهي أن يتم بطريقة مباشرة دون تعقيد لثنائية العلامة.
4. في حالة الجمع إذا كان هناك **واحد زائد** بعد ثنائية العلامة **يضاف في أقصى اليمين مرة أخرى**.

3. الإكمال الثنائي

1. تقم بنفس عمل خوارزمية الإتمام الأحادي في **تحويل قيمة الثنائية** ثم **يضاف واحد في أقصى اليمين للرقم السالب** قبل عملية الجمع.

نلاحظ : أ. يأخذ نفس القيمة.
ب. يتم بطريقة عادية دون أي اعتبار لثنائية العلامة.

لا تظلمن إذا ما كنت مقتدراً فالظلم ترجع عقباه إلى الندم
تنام عيناك والمظلوم منتبهاً يدعوا عليك الله وعين الله لم تنم

مثال (1) : مستخدماً الخوارزميات الثلاثة أجمع **موجب 13 مع سالب 11.**

- **علامة**

	0	0	0	0	1	1	0	1	=	13 +
	1	0	0	0	1	0	1	1	=	11 -
	1	0	0	1	1	0	0	0	=	3 +

ثنائية العلامة
)

- **الإكمال الأحادي**

	0	0	0	0	1	1	0	1	=	13 +
	1	1	1	1	0	1	0	0	=	11 -
	0									
								1		
	0	0	0	0	0	0	1	0	=	2+

ينقل الواحد من الخانة الزائدة إلى أقصى اليمين ثم يجمع مرة أخرى ليعطي النتيجة الصحيحة.

- **الإكمال الثنائي**

	0	0	0	0	1	1	0	1	=	13 +
	1	1	1	1	0	1	0	1	=	11 -
	0	0	0	0	0	0	1	0	=	2 +

الواحد الذائد هنا يُهمل ولا يكتب لأنه تمت إضافته مسبقاً للرقم

ثانياً : تمثيل الأعداد الحقيقية

تخزن الأعداد الحقيقية في جزءين من الكلمة.

1. **الجزء الأول** : يخزن فيه الكسر ويسمى .

(**المانتيسا /الخانة الكسرية المؤثرة:** هي الجزء الكسري من الكلمة).

2. **الجزء الثاني** : تخزن فيه **القوى** **الأس**.

العدد الحقيقي = ضرب الجزء الأول × الجزء الثاني مرفوعاً للقوى.

- **تطبيع الكسر:**

➢ هو أن يكون الرقم اليمين الخانة العشرية أكبر من صفر، والرقم الواقع شمال الخانة العشرية يساوي .

،

- عندما تتحرك الفاصلة العشرية في اتجاه **اليمين** " **سالبة**".
- تتحرك الفاصلة العشرية في اتجاه **اليسار** " **موجبة**".

" " ← ، → " "

- **أمثلة**

1) 1000 0.1×10^{4}

2) 75000 0.75×10^{5}

3) 0.00035 0.35×10^{-3}

5. تحديد عدد الثنائيات لكل جزء ( ).

مثال (1) : إذا كان لدينا حاسوب طول كلمته 40 ثنائية يمكننا تقسيم كلمته كما يلي لتخزين العدد الحقيقي:

الكسر / المانتيسا	علامة الكسر	القوة	علامة القوة
32 ثنائية	1 ثنائية	6 ثنائية	1 ثنائية

مثال (2) : مثل العدد الحقيقي 215678.35 في حاسوب طول كلمته 18 ثنائية وحدد المانتيسا والأس؟

1. **تطبيع الكسر** 215678.35 = 0.21567835×10^{6}
2. **في حاسوب طول كلمته 18 ثنائية** تحسب كما يلي :

المانتيسا	+	الأس	+
10	1	6	1

3. **المانتيسا** = **0.21567835**
4. **الأس** = **6**

مثال (3) : مثل العدد الحقيقي 0.0000587 في حاسوب طول كلمته 32 ثنائية وحدد المانتيسا الأس.

1. **تطبيع الكسر** 0.0000587 = 0.587×10^{-4}
2. في حاسوب طول كلمته 32 تحسب كما يلي :

المانتيسا	+	الأس	-
24	1	6	1

3. **المانتيسا** = **0.587**
4. **الأس** = **- 4**

خوارزمية جمع الأعداد الحقيقية

عند جمع الأعداد الحقيقية ، هناك عدة خطوات يجب إتباعها وهي :

1. تطبيع الكسر.
2. توحيد الأسس وذلك بتحرك الفاصلة العشرية للعدد الأصغر نحو الشمال حتى تصبح قوة العدد الأصغر مثل قوة العدد الأكبر.
3. استخراج عامل مشترك ثم أجمع الكسور.
4. طبع الناتج إلى عدد الخانات العشرية المطلوبة.

مثال (4) : : أجري العملية 750 + 0.0035 ، على حاسوب يتيح 6 خانات ، وأخر يتيح 8 خانات في الجزء الكسري ثم حدد المانتيسا والأس ثم علق على النتائج ، وأخيراً مثل الناتج في حاسوب طول كلمته 24 ثنائية .

1. يـ $0.75 \times 10^{3} + 0.35 \times 10^{-2} =$
2. توحيد الأسس وذلك بتحرك العلامة العشرية للعدد

$0.35 \times 10^{-2} = 0.0000035 \times 10^{3}$

.3 $10^3 \times (0.75 + 0.0000035)$ =

$\underline{\mathbf{10^3 \times 0.7500035}}$ =

.4 6 خانات عشرية = $\underline{\mathbf{10^3 \times 0.75004}}$

.5 8 خانات عشرية = $\underline{\mathbf{10^3 \times 0.7500350}}$

.6 المانتيسا (نتيجة) = $\underline{\mathbf{0.7500035}}$

.7 = $\underline{3}$

.8 الحاسوب الذي يتيح 8 ثنائيات في الجزء الكسري له سعة تخذينية أكبر ويعتبر أحدث من الحاسوب الذي يتيح 6 .

.9 في حاسوب طول كلمته 24 ثنائية تحسب كما يلي :

1	6	1	16
+	3	+	0.7500035

ثالثاً : تمثيل الحروف

هناك نظامان لتمثيل الحروف (أنظر الجداول الملحقة أخر هذا الباب)

النظام الأول – آسكي ASCII	النظام الثاني – أبسيدك EBCDIC
النظام القياسي	نظام شركة آي. بي. أم. IBM
يبدأ بترميز الأرقام ثم الحروف الكبيرة ثم الحروف الصغيرة.	يبدأ بترميز الحروف الصغيرة ثم الكبيرة ثم الأرقام .

رابعاً : تمثيل البيانات المنطقية

يتم تمثيلها **بثنائية واحدة** فقط، وهي

- عندما تكون قيمة هذه الثنائية **واحداً** **صحيح حقيقة.**
- عندما تكون قيمة هذه الثنائية **صفر** **خطا غير حقيقة.**

التحول من نوع بيانات لآخر

تتيح جميع لغات البرمجة التحول من نوع بيانات لآخر كما يلي :-

1. تحويل الأعداد الرقمية إلى حقيقية

تضاف لها علامة عشرية باستخدام عملية التطبيع ، **مثال :**

أ. **28** = 0.28 × 10^2

ب. **512** = 0.512 × 10^3 .

2. تحويل الأعداد الحقيقية إلى رقمية

يتم تحويل الأعداد الحقيقة إلى رقمية إما بالكشط أو التقريب ، مثال حول العدد 329.7 **إلى رقم**

أ. **بالتقريب** = 330

ب. **بالكشط** = 329

☻ تحويل الحروف إلى أرقام

)

(

- **مثال :** إذا " " يُمثل ثنائياً بـ " 100000001" يتم تـ يـ

2^8	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
1	0	0	0	0	0	0	0	1

$= 2^8 + 2^0 = 256 + 1 = 257$.

السجلات والمصفوفات

_____ : يتكون من مجموعة من الحقول التي تشترك في وصف شيء واحد.

1. يتكون من **بعد واحد** ( ).
2. **لا يشترط** **بياناته من نوع واحد** (رقمية وحقيقية ومنطقية و حرفيه).

- يتم تخزين **حقول** **مواقع متجاورة بالذاكرة**.
- **حقل** داخل السجل يمثل **وحدة بيانية مفردة** (متغير واحد).

الرقم	الاسم	العنوان	العمر	التلفون	التقدير
1	إبراهيم طه	الخرطوم	22	787984	جيد
2	عبدالله الأمين	أم درمان	17	554789	ممتاز
3	يس الصادق	بحري	16	325486	نجاح

_____ : عبارة عن بنائية بيانية مركبة ، (أو مجموعة من الوحدات البيانية التي تشترك في وصف شيء واحد).

والمصفوفات لها المواصفات التالية :

1. **أبعاد مركبة** ( )، وكل بعد مركب يتكون من عدد من الوحدات البنائية.
2. كل وحداتها البنائية من **نوع واحد**، (حرفية أو حقيقية أو رقمية أو منطقية).

- يتم تخزين **السجلات** **مواقع متجاورة في الذاكرة**.
- **سجل** داخل المصفوفة يمثل **وحدة بيانية مفردة**.

مثال : **طالب** فإننا نقصد أننا نريد أن نعرف أسمه، وتاريخ ميلاده، والسنة الدراسية التي يدرس فيها، وترتيبه، ودرجاته، وسلوكه، وعنوانه، ورقم تلفونه ،... الخ، كل هذه الوحدات البيانية تشترك في وصف أو إعطاء معلومات عن ذلك الطالب – إذن هناك متغير أساسي (متغير بسيط / واحد) هو **الطالب** متغيرات مفردة وهي أوصاف الطالب، وكل مفردة من هذه المفردات التي ترتبط بوصف الطالب تكون بنائية بيانية مركبة.

أ (س)
مصفوفة ذات بعد واحد
البعد = 1 ( )
طول البعد = 5 (=)
النوع البيانات = حرفية

محمد على يس
عمر الصافي طه
أحمد فضل الله
إبراهيم يوسف
يعقوب الفاضل

مصفوفة أسماء الطلاب

أ (س ، ص)
مصفوفة ذات بعدين
البعد الأول = 5 = س
البعد الثاني = 9 = ص
النوع = رقمية

الترتيب		إنجليزي	رياضيات	إسلامية	فيزياء	أحياء	كيمياء
4	655	95	93	92	95	96	91	93
2	598	91	88	86	87	83	82	81
1	674	96	98	97	96	97	94	96
3	664	98	94	93	96	97	92	94
5	631	86	91	90	92	93	89	90

مصفوفة درجات الطلاب

أ (س ، ص ، ع)
= 5 ( ) =
= 9 ( ) =
= 1 ( ) =
نوع البيانات = منطقية

التقدير

ممتاز
جيد جداً
ممتاز
جيد جداً
جيد

مصفوفة تقدير الطلاب

- **من مثال المصفوفة أعلاه حدد الآتي :**

1. (3).
2. تقدير الطلاب (5 8 5)
3. (6 3)
4. (5) (1).

السؤال	المعنى	الحل
أسماء الطلاب (3)	3 .	أحمد فضل الله
درجات الطلاب (6،3)	3 .6	96
تقدير الطلاب (5،8،5)	5 ودرجته الواقعة في العمود رقم 8 . وتقديره في الصف الخامس من مصفوفة التقدير.	يعقوب **الفاضل** ودرجته في الأحياء **89** وتقديره في مصفوفة التقدير **جيد**.
أسماء الطلاب (5)، الاسم (1)	5 .	ي

- **تمرين : حول السجل السابق إلى مصفوفة وحول المصفوفة أعلاه إلى سجل.**

** طريقة إيجاد موقع عنصر من عناصر المصفوفة في الذاكرة **

1. إذا كانت المصفوفة ذات بعد واحد

موقع العنصر أ (س) = الموقع الأساس + (رقم العنصر – 1)

مثال (1): أوجد موقع العنصر إذا كان الموقع الأساس يساوي 253 والعنصر رقم 3.

أ (س) = 253 + (3 – 1)
= 253 + 2 = 255

2. إذا كانت المصفوفة ذات بعدان

موقع العنصر أ(س، ص) = الموقع الأساس + (ص – 1) × البعد الأول + (س – 1)

مثال (2) : أوجد موقع العنصر في مصفوفة ذات بعدين طول البعد الأول يساوي 15 والبعد الثاني يساوي 9، إذا كان الموقع الأساس يساوي 253 والعنصر رقم (6 ، 7).

أ (6 ، 7) = 253 + (7 – 1) × 15 + (6 – 1)
= 253 + 6 × 15 + 5 = 348

مثال (3) : إذا كان موقع العنصر في مصفوفة هو 210 ، وكانت أبعادها (4 × 5) ، ورقم عنصرها (3 ، 3) ، أوجد كل من :

1. الموقع الأساس .
2. مجموع المواقع التخزين التي تحتاجها المصفوفة لتخزين عناصرها بالذاكرة.

الحل

1. $(1 - 3) + 4 \times (1 - 3) + = 210$

$2 + 4 \times 2 + = 210$

$200 = 10 - 210$

$= 200$

2. مجموع مواقع التخزين = عدد السجلات × عدد المواقع

$4 \times 5 = 20$

(1،5) 216	(1،4) 212	(1،3) 208	(1،2) 204	(1،1) 200
(2،5) 217	(2،4) 213	(2،3) 209	(2،2) 205	(2،1) 201
(3،5) 218	(3،4) 214	(3،3) 210	(3،2) 206	(3،1) 202
(4،5) 219	(4،4) 215	(4،3) 211	(4،2) 207	(4،1) 203

تصنيف العناوين والبيانات والمتغيرات والثوابت داخل الذاكرة

- يمكن تصنيف **عناوين البيانات** داخل الذاكرة إلى صنفين

عناوين البيانات الثابتة	عناوين البيانات المتغيرة
عناوين بيانات لا يمكن تغيرها، ومواقعها لا تتقبل إدخال بيانات جديدة. **مثال** (عناوين).	عناوين بيانات يمكن تغيرها أو تتغير وفقاً - **مثال** (في البنك أو كمية بضاعة داخل مخزن).

- وكذلك يمكن تصنيف **نوع العناوين** إلى قسمين بسيط ومركب

عنوان الذاكرة البسيط	عنوان الذاكرة المركب
هو العنوان الذي يشير إلى موقع واحد . (" " أو متغيراً " ").	هو العنوان الذي يشير إلى أكثر من مثال (مثل عناوين).

- **وكما يمكن تصنيف نوع المتغيرات في الذاكرة إلى قسمين متغير بسيط ومتغير مركب.**

المتغير البياني البسيط	المتغير البياني المركب
هو المتغير الذي يعبر عن وحدة بيانية واحدة ( ) - (رياضة ..).	هو المتغير الذي يعبر عن أكثر من وحدة بيانية واحدة – ( ).

☻ **كيف يتم تخزين السجلات داخل الذاكرة ؟**

يتم تخزين السجلات داخل الذاكرة بنفس طريقة تخزين المتغيرات البسيطة (مثل:).

☻ **كيف يتم الوصول لعناصر السجلات داخل الذاكرة ؟**

يتم عن طريق **أسماء الحقول عناوين للحقول.**

☻ **كيف يتم تخزين المصفوفة داخل الذاكرة؟**

عند تخزين أي مصفوفة بالذاكرة هنالك عدد من الخطوات يقوم بها الكمبيوتر لتخزين المصفوفة وهي :

1. . (هو موقع بالذاكرة والذي يبدأ منه على التوالي تخزين عناصر المصفوفة).
2. تخزين .
3. تخزين .

☻ **كيف يتم الوصول لعناصر المصفوفة داخل الذاكرة ؟**

يتم الوصول لعناصر المصفوفة في الذاكرة عن طريق :

أ. مؤشر العنصر. ب. طول البعد. ج. الموقع الأساس.

لغة باسكال

هي واحدة من لغات البرمجة العالية المستوى، وتتميز عن بقية اللغات في أنها لغة تعليمية وبرمجية بسيطة وسهلة التعلم والقراءة وتعتمد منهج الصرامة في الأسلوب مع التأكيد على قوتها كلغة برمجة متكاملة وتستخدم لكتابة برامج مرنة وفعالة.

الشكل العام للبرنامج في لغة باسكال

الجزء الأول رأس البرنامج	دائماً يبدأ بكلمة " برنامج Program " أسم البرنامج (أدخال ، أخراج) مثال : برنامج مرتبات الموظفين (إدخال ، إخراج)؛ Program Emp_Salaries (I,O);
الجزء الثاني جزء التعريفات والإعلانات	يتم فيه الإعلان عن كل الجمل المستخدمة مثل الثوابت العناوين التي لا تقبل تغير قيمتها في Const: المتغيرات تتغير وفقاً لمتطلبات البرنامج ونفصل بينهما بشولة المستخدمة بواسطة كلمة:Var التعريفات الجديدة Type : الدوال USES: الدوال Function : اللافتات Label: الإجراءات Procedure:
الجزء الثالث جسم البرنامج	هو الجزء الأساسي ودائماً يبدأ بكلمة أبدأ Begin { .... } وينتهي البرنامج بكلمة نهاية . End .

مثال 1: أكتب برنامج بلغة باسكال يقوم بحساب مساحة الدائرة التي نصف قطرها r مساحتها تحسب وفقاً للمعادلة التالية :

area = pi * r * r

```
Program area ( input,output);

Const  pi = 3.14;
Var
        Area, r : real;
Begin
        Area = pi*i *i;
Write ('أدخل قيمة نصف القطر');
Readln (r);
Writeln ('مساحة الدائرة =' area);
End.
```

```
برنامج المساحة (إدخال،إخراج) ؛
الثوابت  بآي = 3.14؛
متغيرات
        مساحة، نق: حقيقي؛
بداية
        Area = pi* r*r ؛
أكتب ( 'أدخل قيمة نصف القطر' )؛
أقرأ (r) ؛
أكتب ( ' مساحة الدائرة = ' Area )؛
نهاية.
```

مثال 2: أكتب برنامج بلغة باسكال يقوم بحساب مساحة أي مثلث قائم الزاوية ارتفاعه a وقاعدته b علماً بأن مساحة المثلث تحسب وفقاً للمعادلة التالية : **area = ½ a.b**

برنامج المساحة (إدخال،إخراج) ؛
متغيرات
area ، a ، b ، : حقيقي؛
بداية
area = ½ a.b ؛
أكتب ('أدخل قيمة طول الارتفاع')؛
أقرأ (a) ؛
أكتب ('أدخل قيمة طول القاعدة')؛
أقرأ (b) ؛
أكتب ('مساحة المثلث = ' area) ؛
نهاية.

```
Program area ( I ,o );

Var
        Area, a , b : real;
Begin
        Area = a*b /2;
Write  ('أدخل قيمة طول الإرتفاع');
Readln (a);
Write  ('أدخل قيمة طول القاعدة');
Readln (b);
Writeln ('مساحة المثلث =' area);
End.
```

مثال 3 : أكتب برنامجاً يشبه باسكال يقوم بحساب مساحة أي مثلث أضلاعه a،b،c – علماً بأن مساحة المثلث تحسب وفقاً للمعادلة التالية:

$$Area = s(s-a)(s-b)(s-c) \quad since \quad s = (a+b+c)$$

برنامج المساحة (ادخل ، اخراج)؛
متغيرات
a b c s area : حقيقي؛
أبدأ
area = s(s – a)(s – b)(s – c)؛
s = (a + b + c) ؛
أكتب ('أدخل قيمة الضلع الأول ')؛
أقرأ (a) ؛
أكتب ('أدخل قيمة الضلع الثاني ')؛
أقرأ (b) ؛
أكتب ('أدخل قيمة الضلع الثالث ')؛
أقرأ (c) ؛
أكتب ('مساحة المثلث = ' ، area) ؛
النهاية .

```
Program Area ( I , O );
  Var
    a, b, c, area : real;
Begin
    area  =  s( s – a)(s – b)(s – c);
    s  =  (a + b + c)  ;
    Writeln (' input  value of   a ');
    Read ( a );
    Writeln (' input  value of   b ');
    Read ( b );
    Writeln (' input  value of   c ');
    Read ( c );
    Writeln ('the  area  is  = ' , area);
End.
```

var : var name : data type	Const = const name = const value

☻ الوحدات

هناك العديد من الوحدات التي توفرها لغة باسكال للمستخدم، وتحتوي هذه بداخلها على العديد من جزء التعريفات والإعلانات يتم أستخدام هذه هذه الوحدات **تنشيطها**

أولاً الكلمة المحجوزة USES تليها أسم الوحدة المراد تنشيطها.

مثال:

أسم الوحدة المراد أستخدامها uses

Uses **dos**
Uses **graph**
Uses **crt**

☻ تعريف المتغيرات في لغات البرمجة

لغة بيسك :	لا تشترط تعريف نوع البيانات في المتغيرات وهذا له عدة مشاكل.
لغة فورتران :	تتعرف تلقائياً على نوع المتغير حسب الحرف الأول المستخدم، فإذا كان الحرف الأول بين **I** **N** فإنها تعتبر المتغير عدداً رقمياً تعتبر عدداً حقيقياً.
لغة سي وباسكال :	بتعريف المتغيرات في أول البرنامج في جزء التعريفات والإعلانات.

عبارات الإدخال والإخراج في لغة باسكال

☻ عبارتي الإدخال

تستخدم لغة باسكال العبارتين **Read** و **ReadLn** لقراءة المعلومات من لوحة المفاتيح .

- **مثال 1 :** إذا كان لدينا سطرين من الأرقام، السطر الأول يحتوي على الأرقام التالية 10 20 30 40 50 60 70 80. وطلبنا من البرنامج قراءة هذان السطران **Read** **ReadLn** فإن النتيجة ستكون كما يلي :

Read	ReadLn
10 , 20 , 30 , 40 **50** , 60 , 70 , 80 **read**(numbers, **A,B,C**); **read** (numbers, **D,E**); النتيجة (تتم القراءة على التوالي) **10,20,30,40,50**	**10 , 20 , 30** , 40 **50 , 60** , 70 , 80 **readln** (numbers, **A,B,C**); **readln** (numbers, **D,E**); النتيجة (تتم القراءة على التوازي) **10,20,30,50,60**

☻ عبارتي الإخراج

تستخدم لغة باسكال العبارتين **Write** و **WriteLn**
على ملف محدد، والفرق الوحيد بينهما أن عبارة WriteLn تنقل المؤشر إلى سطر جديد بعد إظهار أو كتابة المعلومات بينما عبارة write لا تنقل المؤشر لسطر جديد.

- **مثال2 : إذا كان لدينا الاسم Sudan Republic فيمكننا طباعته كما يلي :**

WriteLn	Write
Writeln ('Sudan'); Writeln ('Republic'); النتيجة (تتم الكتابة على التوازي) Sudan Republic	Write ('Sudan'); Write ('Republic'); النتيجة (تتم الكتابة على التوالي) Sudan Republic

- **القاعدة لتعريف السجلات**

Type record name = record; .... } Record fields End;	نوع أسم السجل = سجل { .... نهاية ؛

- **مثال3** أكتب شفرة توضح كيفية الإعلان عن ـــــ ى عناوين address ويحتوي على البيانات التالية:

الأسم30 حرف ، الشارع25 حرف ، المدينة 15حرف ، الرمز رقمي.

Type address = record Names : array (1 … 30) of char ; Street : array (1 … 25) of char ; City : array (1 … 15) of char ; Code : integer ; End ;	نوع عناوين = سجل : (1 ... 30) حرفية ; : (1 ... 25) حرفية ; المدينة : (1 ... 15) حرفية ; : ; نهاية;

- **القاعدة لتعريف المصفوفة**

Var Array name : array (array dimension) data type;	متغير : ( ) نوع بيانات المصفوفة

- **مثال 4 : أعلن عن مصفوفة أسمها أسماء السلع items names طولها 60 .**

متغيرات حرفية (60 ... 1) :	**Var** Items Price : Array (1 …60) of char ;

- **مثال 5 : أعلن عن مصفوفة أسمها الطلاب Students عناصرها درجات الطلاب تتكون من 4 صفوف و5 أعمدة.**

متغيرات رقمية (5... 1 4 ... 1) :	**Var** Students : Array (1 … 4 , 1 … 5) **int;**

وأعمل لطاعته تنل منه الرضا ... إن المطيع لربه لمُقرّب

فأقنع ففي بعض القناعة راحة ... واليأس مما فات فهو المطلب

وإذا طعمت كُسيت ثوب مذلةٍ ... فلقد كُسي ثوب الذلة أشعب

وتوق من غدر النساء خيانة ... فجميعهن مكائد لك تُنصب

والقى عدوك بالتحية لا تكن ... منه زمانك خائفاً تترقب

وأحذره يوماً إذا أتى لك باسماً ... فالليث يبدو نابه إذ يغضب

إن الحقود وإن تقادم عهده ... فالحقد باقٍ في الصدور مغيب

وإذا الصديق رأيته متملقاً ... فهو العدو وحقه يُتجنب

لا خير في ودّ أمرىء متملق ... حلو اللسان وقلبه يتلهب

يلقاك يحلف أنه بك واثق ... وإذا توارى عنك فهو العقرب

يعطيك من طرف اللسان حلاوة ... ويروغ منك كما يروغ الثعلب

وأختبر قرينك وأصطفيه تفاخراً ... إن القرين إلى المقارن يُنسب

.

الحلقة FOR.....DO .

تستخدم هذه الحلقة إذا كان معروف لدينا مسبقاً عدد المرات التي نريد فيها تكرار الأمر أو العمل.

الصيغة العامة

For < متغير > := < قيمة المتغير الابتدائية > to [or downto
< قيمة المتغير النهائية > do
< عبارات بسيطة أو مركبة >

استخدام العبارات البسيطة (يستخدم فيها متغير واحد)

مثال 1 : أكتب برنامج يقوم بحساب مربع الأعداد من 5 إلى 10.

```
Var
      I : integer;                          جعل المتغير I متغير رقمي
  Begin
      For I : = 5 to 10 do;                 يحسب كل الأرقام الواقعة بين 5 10 وجعلها قيمة للمتغير I
      Writeln ( I * I );                    يقوم بضرب أي رقم بين ال 5 10 في نفسه ثم طباعته
  End.
```

مثال 2 : برنامج يقوم بحساب مربع الأعداد من 10 إلى 5

```
Var
      I : integer;                          جعل المتغير I متغير رقمي
Begin
      For I : = 10 downto 5 do;             يحسب كل الأرقام الواقعة بين 10 5 وجعلها قيمة للمتغير I
      Writeln ( I * I );                    يقوم بضرب أي رقم بين ال 10 5 في نفسه ثم طباعته
End.
```

العبارات المركبة (يستخدم فيها أكثر من متغير)

تتكون من عدد من العبارات البسيطة ويجب أن تنحصر بين العبارتين Begin.....End .

مثال 3 : أكتب برنامج يقوم بحساب مربعات الأعداد من 10 إلى 5

```
Var
      I , X : integer;                      إعتبار المتغير I X متغيرات رقمية
Begin
      For I : = 10 downto 5 do;             يحسب كل الأرقام الواقعة بين 10 5 وجعلها قيمة للمتغير I
          Begin
          X : = I * I ;                     جعل ناتج قيم المتغير I في نفسه تساوي X
          Writeln ( x );                    طباعة قيمة المتغير X الأرقام بين 10 5
          End;
End.
```

مثال 3 : أكتب برنامج يقوم بحساب الزوجية 5 10 ثم إيجاد الوسط الحسابي لها.

```
Program FOUR;
    Var
        I , S : integer; ;                    إعتبار المتغير I S متغيرات رقمية
    Begin
        S = 0;                                جعل قيمة S الإبتدائية
        For I : = 5 to 10 do;                 يحسب كل الأرقام الواقعة بين 5 10 وجعلها قيمة للمتغير I
        Begin
            S : = S + I * I ;                 أجعل قيمة S النهائية = قيمة S الإبتدائية + I × I
            Writeln ( S , S/6 );              أطبع قيمة كل من S S/6
        End
    End.
```

مثال 4 : أكتب برنامج يقوم بحساب مجموع مربعات الأعداد من a إلى b ، حيث a أكبر من b ، ثم إيجاد الوسط الحسابي .

```
Program FIVE;
    Var
        I , a, b, s : integer; ;              إعتبار المتغير I a b s متغيرات رقمية
        Avg : real;                           avg متغير حقيقي
    Begin
        Writeln ('A , B أدخل قيمة ') ;
        Realln ( a , b);                      قراءة قيمة المتغيرات a b
        S: = 0;                               جعل قيمة S البدائية تساوي صفر
          For I: = a  to  b do;               يحسب كل الأرقام الواقعة بين a b وجعلها قيمة للمتغير I
          S:  = S + I* I ;                    أجعل قيمة S النهائية = قيمة S الإبتدائية + I × I
         Avg: =S / (a- b + 1);                حساب الوسط الحسابي لكل القيم الواقعة بين a b
      Writeln (' =                  ' , S);   طباعة مجموع مربعات كل الأرقام الواقعة بين a b
      Writeln ( ' =          ' , avg );       طباعة متوسط كل الأرقام الواقعة بين a b
    End.
```

مثال 5 : أكتب برنامج بلغة باسكال يقوم بحساب مضروب العدد N.

```
Var
    I, n : shortint ;
    Fact : longint
Begin
    Writeln ("                  " );
    Readln (n);
    Fact: 1;
    For I: =N down to 1 do
    Fact : fact * I;
    Writeln ("fact = " , fact);
End
```

الحلقة Repeat.....Until

تستخدم هذه الحلقة إذا كان تكرار العمل يتوقف على شرط معين.

الصيغة العامة

```
Repeat
        < عبارات بسيطة أو مركبة >
Until   < تعبير بولياني >
```

مثال 5 : أكتب برنامج لإيجاد مجموع الأعداد الزوجية بين 1 إلى 10

```
Program  SIX ;
    Var
        I , sum : integer ;                 إعتبار المتغير I  sum  متغيرات رقمية
    Begin
        I : = 0 ; sum : = 0 ;               جعل قيمة S  sum  البدائية تساوي صفر
        Repeat                              أبتداء الحلقة التكرارية
            I : =  I + 2;                   أجعل قيمة المتغير I الحالية تساوي قيمة I  2 +
            Sum : = sum + I ;               أجعل الناتج الجديد = الناتج السابق + 1
        Until ( I = 10 ) ;                  توقف عندما قيمة I  10
    End.
```

الحلقة WHILE.....DO

تستخدم إذا كان تكرار العمل أو الأمر يتوقف على شرط معين.

الصيغة العامة

```
While   < تعبير بولياني >  do
        < عبارات بسيطة أو مركبة >
```

مثال 6 : أكتب برنامج لإيجاد مجموع الأعداد الزوجية بين 1 إلى 10.

```
Program SEVEN ;
    Var
        I , sum : int ;                     إعتبار المتغيرات I  SUM  متغيرات رقمية
    Begin
        I : = 0 ; sum : 0 ;                 جعل قيمة S  sum  البدائية تساوي صفر
        Begin
            While ( I = 10 ) do;            إعتبار أعلى قيمة للمتغير  تساوي 10
            I : = I + 2;                    أجعل قيمة المتغير I الحالية تساوي قيمة I  2 +
            sum : = sum + I                 أجعل الناتج الجديد = الناتج البدائية + 1
        End ;
    End .
```

العبارة IF…….THEN…… ELSE

تستخدم هذه العبارة للاختيار بين خيارين

الصيغة العامة

```
If  < تعبير بولياني >  then
      < عبارات بسيطة أو مركبة >
else
      < عبارات بسيطة أو مركبة >
```

تعبير بولياني يعني تعبير

مثال 7 : أكتب برنامج يقوم بطباعة عبارة "نجاح" إذا كانت درجة الطالب أكبر من أو تساوي 50 وإلا يطبع " رسوب"

```
Program EIGHT;
    Var
        Score : shortint ;                          إعتبار المتغير score متغيرات رقمية
    Begin
        Write ( 'أدخل درجة الطالب' ) ;              يطلب البرنامج من المستخدم إدخال درجة الطالب
        Readln ( score );                           تعويض قيمة المتغير score
        If score >= 50 then                 50
        Writeln ( ' نجاح ' );                       نجاح
       Else                                         غير ذلك
        Writeln ('رسوب  ');                         رسوب
    End.
```

مثال 8 : أكتب برنامج يقوم بإدخال أي حرف، وإذا كان الحرف المُدخل أحد حروف العلة (A, E, I, O, U) ، يكتب " حرف علة " وإلا ، يكتب " ليس حرف علة ".

```
Program NINE;
    Var
        C : char; ;                                 إعتبار المتغير C متغير حرفي
    Begin
        Writeln ( ' أدخل أي حرف' );                 يطلب البرنامج من المستخدم إدخال أي حرف
        Readln ( C );                               تعويض قيمة المتغير c
        C: = upcase ( c );                          معاملة الحروف الكبيرة والصغيرة على أنهما شيء واحد
    If  C  in ['A' , 'E' , 'I', 'O','U']  then      A , E , I , O , U
        Writeln ('حرف علة');                        حرف علة
          Else                                      غير ذلك
           Writeln ('ليس حرف علة' );                ليس حرف علة

    End.
```

مثال 9 : أكتب برنامج يقوم بطباعة " كسلا " أو " جوبا " أو " مدني " أو " الفاشر" أو " دنقلا" ، إذا كان الرقم المدخل هو 1 أو 2 أو 3 أو 4 أو 5 على الترتيب.

```
Var
    X : char;                                   إعتبار المتغير X متغير حرفي
Begin
    Writeln (' أدخل رقم من 1 إلى 5 ');          يطلب البرنامج إدخال أي رقم بين 1 5
    Readln (x);                                 تعويض قيمة المتغير x
  If not X in [1…5] then                        إذا لم تكن قيمة المتغير x بين 1 5
    Writeln ('الرقم المدخل خارج المدى المحدد');  الرقم المدخل خارج المدى المحدد
  Else                                          غير ذلك
  Begin
    If (x = 1) then writeln (' كسلا ');         إذا كان الرقم المدخل يساوي 1 كسلا
    If (x = 2) then writeln (' جوبا ');         إذا كان الرقم المدخل يساوي 2 جوبا
    If (x = 3) then writeln (' مدني ');         إذا كان الرقم المدخل يساوي 3 مدني
    If (x = 4) then writeln (' الفاشر ');       إذا كان الرقم المدخل يساوي 4 الفاشر
    If (x = 5) then writeln (' دنقلا ');        إذا كان الرقم المدخل يساوي 5 دنقلا
  End;
End.
```

معالجات الحاسوب لأنواع البيانات المختلفة في لغة باسكال

هناك معالجات تتفق فيها كل لغات البرمجة في التعامل مع أنواع البيانات المختلفة وهي :

☻ معالجات الأعداد الرقمية

الضرب	*	ضرب الأرقام الصحيحة ينتج رقماً	25 = 5×5.
القسمة	/	ج عدد حقيقية أو رقم.	5= 2÷10 3.5 = 3÷10
الجمع	+	ينتج رقماً.	10 = 5+5
الطرح	-	ينتج رقماً.	5= 5 – 10
القسمة	y div x	هذه قسمة نتج رقماً	. X Y 1 = 4÷ 7 Y=7 , X = 4
باقي القسمة	y mod x	y نتج رقماً هو باقي قسمة Y . X	3= 4÷7
الجزر التربيعي	sqr (x)	نتج رقماً X .	3 = √9
الكشط	trunc (x)	نتج رقماً من العدد الحقيقي X .	2 = 2.6
التقريب	round (x)	نتج رقماً من العدد الحقيقي X بتقريب الكسر.	3 = 2.6

معالجات الأعداد الحقيقية

	.	عدد حقيقي × عدد حقيقي أو × رقم يكون	الناتج عدداً حقيقياً.
	/	عدد حقيقي ÷ عدد حقيقي أو ÷ رقم يكون	الناتج عدداً حقيقياً.
	+	عدد حقيقي + عدد حقيقي أو + رقم يكون	الناتج عدداً حقيقياً.
	-	عدد حقيقي + عدد حقيقي أو + رقم يكون	الناتج عدداً حقيقياً.
القيمة المطلقة	abs (x)	القيمة المطلقة	لا تغير من نوع العدد.
الجزر التربيعي	sqr (x)	المربع لأي عدد حقيقي	ينتج عدداً حقيقي.
	sin (x)	هذه دوال	تنتج عدداً حقيقياً
	cos (x)		
	tan (x)		
	arctan(x)		
اللوغريثم الطبيعي	Ln (x)	سواء أكانت قيمة X **رقماً** أم **عدداً**	
	exp (x)		

معالجات البيانات المنطقية

And		ينتج قيمة منطقية.
or		ينتج قيمة منطقية.
not		نتج قيمة منطقية.
>	أقل من ينتج منها	نتج قيمة منطقية.
=>	أقل من أو يساوي	ينتج قيمة منطقية
=	يساوي ينتج منها	نتج قيمة منطقية.
<>	لا يساوي ينتج منها	نتج قيمة منطقية.
odd(x)	تحقيق فردية X	ينتج قيمة منطقية.
eol(f)	تحقيق انتهاء السطر End of Line	ينتج قيمة منطقية.
eof(f)	تحقيق انتهاء الملف End of File	ينتج قيمة منطقية.

☻ معالجات الحروف

الحروف تعني الحروف الهجائية والأرقام وحروف الفراغ.
وهناك دالتان تتعامل مع البيانات الحرفية وهما :-

- **الداله الأولى : Ord (c)** : تستخدم لإيجاد الرقم المقابل للحرف C الترميز و_____ .

مثال : Ord (a) - 65 = 1000001 = a

65 = ord(a)

- **الداله الثانية :** **char (i)** : تستخدم لإيجاد الحرف المقابل للرقم المخزن . المتغير I جدول الترميز _____.

مثال : char (65) - a = 1000001 = 65

a = char (65)

- إذن : الدالة Ord (c) والدالة char (i) متعاكستان

لا تسألن بُني آدم حاجة وسَل الّذي أبَوابهُ لا تُحَجبُ
اللّه يغضب إن تَركت سُؤاله وترى إبن آدم حين يُسأل يَغضبُ

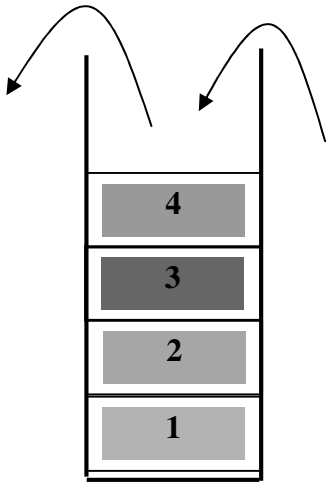
++++++++++++++++++++++

وأصبر على مر الجفا من معلم فإن رسوب العلم من نفراته
ومن لم يذق مر التعلم ساعة تجرع ذل الجهل طول حياته
ومن فاته التعليم وقت شبابه فكبر عليه أربع لوفاته
وذات الفتى والله بالعلم والتقى إن لم يكونا لا إعتبار لذاته

البنائيات المجردة

التعامل مع التطبيقات (البرامج) ( )، لها كفاءة عالية في .

البنائيات المجردة :-



1. المكدسات (المخزنات)

وهذا النظام يعرف لتكديس الأشياء بعضها فوق بعض وليس بالإمكان سحب السفلى قبل العليا، وهذا النظام **يتناسب مع تطبيقات المخازن** .

Last In First Out = LIFO = ليفو

أخر عنصر أضيف أول عنصر يحذف

2. الصفوف

وهذا النظام يعرف . وهذا النظام **يتناسب مع كل التطبيقات التي عادة ما يأخذ فيها الخدمة السابق قبل اللاحق .**

فيفو = FIFO = First In First Out =

أول عنصر أضيف أول عنصر يحذف.

مثال : شباك التذاكر أو صف حجوزات الطيران.

3. القوائم المتصلة

1	إبراهيم	الخرطوم	22	787984	جيد		2	عبدالله	بحري	32	354789	ممتاز		3	عثمان	جبرة	25	365865	نجاح	

تعتبر القوائم المتصلة من أكثر البنائيات المتجردة مرونة

.

الشكل العام للسجل :

	حقول السجل / العقدة				المؤشر

حقل واحد أو أكثر يحوي يانية في السجل

حقل يخزن فيه عنوان وموقع السجل التالي له في الذاكرة

التطبيقات العملية :

يتناسب مع الملفات المتغيرة السجلات التطبيقية : **بيانات عملاء البنك، وحركة حساباتهم المختلفة.**

- **العقدة**

عبارة عن سجل من سجلات القوائم المتصلة، وهناك عقدتان أساسيتان هما :-

العقدة الأولى : هي التي ليس لها عقدة تحفظ عنوانها أو تشير إليها، وتعرف **برأس القائمة**.

العقدة الأخيرة : ليس بها عنوان عقدة أو لا تشير لعقدة تالية لها، وتعرف **بزيل القائمة**.

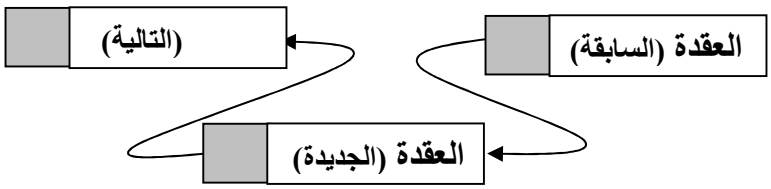
مؤشر الرأس		عقدة 1		عقدة 2		عقدة 3		مؤشر الذيل
	عنوان عقدة الرأس		عنوان العقدة 2		عنوان العقدة 3		عنوان العقدة الوهمية لحفظ الشكل العام للسجل	

المؤشر

هو حقل من حقول السجلات يخزن فيه موقع السجل التالي له في الذاكرة.

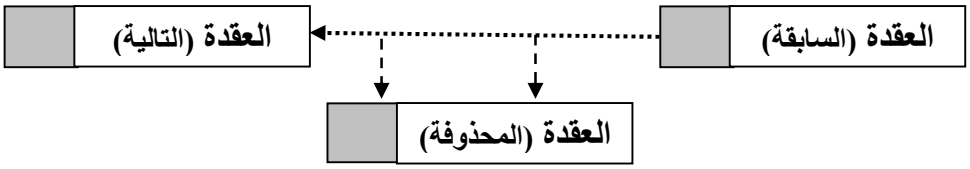
كيفية إضافة عقدة إلى قائمة متصلة

تتم عملية الإضافة بجعل مؤشر العقدة السابقة تشير للعقدة الجديدة ومؤشر العقدة الجديدة يشير للعقدة التالية.



كيفية حذف عقدة من قائمة متصلة

تتم عملية يشير للعقدة التالية.



4. قواعد البيانات

قواعد البيانات من البنائيات المتجردة وهي عبارة عن أنظمة (برامج) يتم تصميمها بواسطة برامج مثل **أس كيو أل SQL أو أوراكل أو فوكس برو**

مثل: .

1. نظام الشرطة :	يشمل البطاقة الشخصية، والجنسية، والجوازات، والحركة ، ودخول وخروج المواطنين الأجانب، وملفات الجرائم والسرقات والشبهات ، ... الخ.
2. نظام الطيران :	يشمل حجوزات المسافرين، والأمتعة، والبضائع، والأمتعة المحظورة والمتخلفة ... .
3. نظام الحسابات :	يشمل حسابات البنوك، وحسابات الشركات، وحسابات الموظفين ورواتبهم، ..

قواعد البيانات العلائقية فهي **ربط منطقي وعملي بين البنائيات المتجردة وقواعد البيانات** ، فمثلاً عند ربط قاعدة بيانات الشرطة مع قاعدة بيانات الطيران مع قاعدة بيانات سابات كلها تمثل بنائية متجردة معقدة تعرف بأسم **قواعد البيانات العلائقية** (قاعدة بيانات متصلة ولها علاقة مع قاعدة بيانات أخرى) **والهدف الأساسي** من هذا الربط هو تبادل المعلومات بين قواعد البيانات المختلفة.

مقارنة بين القوائم المتصلة والمصفوفات

	القوائم المتصلة	المصفوفات
المساحة في الذاكرة	تتقلص أو تزيد كلما حذف أو أضيف عنصر.	لا يمكن زيادة أو نقصان المساحة التي تم تعريفها في بداية البرنامج.
الحذف والإضافة	سهولة الحذف والإضافة.	.
طريقة البحث	تستخدم فقط طريقة البحث المتتالي ويستغرق هذا وقتاً طويلاً.	ويمتاز بالسهولة والسرعة

خوارزمية الإضافة والحذف

يوجد نوعان من الخوارزميات
الذاكرة، والمثال التالي يوضح الكيفية التي تتم بها إضافة أو حذف عنصر.

.	.
هل الموقع الأعلى **يساوي** "**نعم**" لا يمكن الإضافة وتنتهي الخوارزمية. : ( ). "**لا**" تستمر الخوارزمية في عملية الإضافة.	هل الموقع الأعلى **يساوي صفر.** "**نعم**" لا يمكن الحذف وتنتهي الخوارزمية. : ( ). "**لا**" تستمر الخوارزمية في عملية الحذف.
= 1 +	= 1 –
( ) = . (أي يضاف العنصر إلى المصفوفة في الموقع الذي رقمه يساوي الرقم الأعلى السابق زائداً واحد).	( ) = . (أي يحذف العنصر الذي رقمه في المصفوف يساوي الموقع الأعلى).

وفي الجهل قبل الموت موت لأهله ... وأجسادهم قبل القبور قبور
وإن أمرءاً لم يحي بالعلم ميت ... وليس له حتى النشر نشور

:

برنامج إضافة عنصر للمكدس (إدخال ، إخراج);

ثابت

100 =

متغيرات

:

;

:

;

: [...1]

;

أبدأ

( );

( );

=

;

( )

;

=

;1 +

( ) =

;

نهاية

:

برنامج حذف عنصر المكدس (إدخال ، إخراج);

ثابت

;100 =

متغيرات

:

;

:

;

: [...1]

;

أبدأ

أدخل قيمة الموقع الأعلى الحالية;

إذا كانت قيمة الأعلى = صفر;

( )

;

;

=

( );

=

;1 –

نهاية

الجدول ٤ - ٦

تمثيل الحروف الابجدية الانجليزية والأرقام والحروف الخاصة رمز إيسيديك

الحرف	تمثيل إيسيديك	الحرف	تمثيل إيسيديك
A	١١٠٠٠٠٠١	0	١١١١٠٠٠٠
B	١١٠٠٠٠١٠	1	١١١١٠٠٠١
C	١١٠٠٠٠١١	2	١١١١٠٠١٠
D	١١٠٠٠١٠٠	3	١١١١٠٠١١
E	١١٠٠٠١٠١	4	١١١١٠١٠٠
F	١١٠٠٠١١٠	5	١١١١٠١٠١
G	١١٠٠٠١١١	6	١١١١٠١١٠
H	١١٠٠١٠٠٠	7	١١١١٠١١١
I	١١٠٠١٠٠١	8	١١١١١٠٠٠
J	١١٠١٠٠٠١	9	١١١١١٠٠١
K	١١٠١٠٠١٠	.	٠١٠٠١٠١١
L	١١٠١٠٠١١	)	٠١٠١١١٠١
M	١١٠١٠١٠٠	،	٠١٠٠١١٠٠
N	١١٠١٠١٠١	+	٠١٠٠١١١٠
O	١١٠١٠١١٠	$	٠١٠١١٠١١
P	١١٠١٠١١١	*	٠١٠١١١٠٠
Q	١١٠١١٠٠٠	:	٠١٠١١١١٠
R	١١٠١١٠٠١	—	٠١١٠٠٠٠٠
S	١١١٠٠٠١٠	/	٠١١١٠٠٠١
T	١١١٠٠٠١١	,	٠١١٠١٠١١
U	١١١٠٠١٠٠	%	٠١١٠١١٠٠
V	١١١٠٠١٠١	=	٠١١١١١١٠
W	١١١٠٠١١٠	@	٠١١١١١٠٠
X	١١١٠٠١١١	:	٠١١١١٠١٠
Y	١١١٠١٠٠٠	،	٠١١٠١١١٠
Z	١١١٠١٠٠١	?	٠١١٠١١١١
BLANK	٠١٠٠٠٠٠٠		

الجدول ٤ - ٧

تمثيل الحروف الابجدية الانجليزية والأرقام والحروف الخاصة رمز أسكي

الحرف	تمثيل اسكي	الحرف	تمثيل أسكي
A	٠١٠٠٠٠٠١	0	٠٠١١٠٠٠٠
B	٠١٠٠٠٠١٠	1	٠٠١١٠٠٠١
C	٠١٠٠٠٠١١	2	٠٠١١٠٠١٠
D	٠١٠٠٠١٠٠	3	٠٠١١٠٠١١
E	٠١٠٠٠١٠١	4	٠٠١١٠١٠٠
F	٠١٠٠٠١١٠	5	٠٠١١٠١٠١
G	٠١٠٠٠١١١	6	٠٠١١٠١١٠
H	٠١٠٠١٠٠٠	7	٠٠١١٠١١١
I	٠١٠٠١٠٠١	8	٠٠١١١٠٠٠
J	٠١٠٠١٠١٠	9	٠٠١١١٠٠١
K	٠١٠٠١٠١١	.	٠٠١٠١١١٠
L	٠١٠٠١١٠٠	)	٠٠١٠١٠٠١
M	٠١٠٠١١٠١	،	٠٠١١١١٠٠
N	٠١٠٠١١١٠	+	٠٠١٠١٠١١
O	٠١٠٠١١١١	$	٠٠١٠٠١٠٠
P	٠١٠١٠٠٠٠	*	٠٠١٠١٠١٠
Q	٠١٠١٠٠٠١	:	٠٠١١١٠١١
R	٠١٠١٠٠١٠	—	٠٠١٠١١٠١
S	٠١٠١٠٠١١	/	٠٠١٠١١١١
T	٠١٠١٠١٠٠	,	٠٠١٠١١٠٠
U	٠١٠١٠١٠١	%	٠٠١٠٠١٠١
V	٠١٠١٠١١٠	=	٠٠١١١١٠١
W	٠١٠١٠١١١	@	٠١٠٠٠٠٠٠
X	٠١٠١١٠٠٠	:	٠٠١١١٠١٠
Y	٠١٠١١٠٠١	،	٠٠١١١١١٠
Z	٠١٠١١٠١٠	?	٠٠١١١١١١
BLANK	٠٠١٠٠٠٠٠		