



الجامعة السورية الخاصة
SYRIAN PRIVATE UNIVERSITY

وحدة متطلبات الجامعة

مهارات الحاسوب
Computer Skills
2017-2018

إعداد

د.م. حسان محمد أحمد

Hassan.Ahmad@spu.edu.sy

نظم العد

(Numbering Systems)

مقدمة

- قبل اختراع الكتابة كان القدماء يقومون بالعد على أصابع اليد فكانت اليد تمثل الأعداد وعند انتهاء الأصابع كانوا يحتاجون إلى شخص آخر ليقوم بالعد فكان الأول يمثل الآحاد و الثاني يمثل العشرات.
- عند اختراع الكتابة اجتهد العلماء لاختراع منظومة أعداد بدلاً من الاصابع.
- كانت هذه الأعداد هي الأعداد الأساسية وهي (من 0 حتى 9) وبإضافة الواحد إلى الصفر يتكون العدد 10.
- اعتمد العلماء على أنّ كل الأعداد بعد 9 مزيج من عددين أو أكثر مثل 10 و 100 و 6735.
- وضع قواعد هذه الأنظمة العالم العربي: الخوارزمي
- تقوم فكرة أي نظام عد على مبدئين أساسيين:
 - أساس النظام (Base) وهو عدد صحيح موجب.
 - عدد رموز أو مفردات هذا النظام.

تعريف نظام العد:

- هو طريقة للتعامل مع رسوم الأرقام للتعبير عن قيمتها وكيفية تطبيق العمليات الحسابية عليها.
- أو
- هو شفرة لها مجموعة من الرموز تستخدم للتعبير عن الأعداد.

أنظمة العد (Numbering systems)

1- النظام العشري (Decimal System)

- يعتبر النظام العشري أكثر أنظمة العد استعمالاً من قبل الإنسان، وقد سمي بالعشري لأنه يتكون من عشرة أرقام هي (0، 1، 2، ...، 9) و التي بدورها تشكل أساس نظام العد العشري.
- بشكل عام، يمكن القول أن **أساس** أي نظام عد يساوي **عدد الأرقام المستعملة لتمثيل الأعداد فيه**، وهو يساوي كذلك **أكبر رقم في النظام مضافاً إليه واحد**.
- تمثل الأعداد في النظام العشري بوساطة قوى الأساس 10 وهذه تسمى بدورها **أوزان خانات العدد**، أي

$$10^n, \dots, 10^3, 10^2, 10^1, 10^0, 10^{-1}, 10^{-2}, 10^{-3}, \dots, 10^{-n}$$

أمثلة: النظام العشري (Decimal System)

				9876	الرقم
ألف	مئات	عشرات	آحاد	=	الرقم
9	8	7	6	=	
10^3	10^2	10^1	10^0	=	أوزان المواقع
1000	100	10	1	=	القيمة
(9×1000) + (8×100) + (7×10) + (6×1) =					قيمة العدد
9000 +	800 +	70	+ 6	=	
9876					=

يمكن كتابة العدد العشري $N=7129.45$ على النحو الآتي:

$$N = 7 \times 10^3 + 1 \times 10^2 + 2 \times 10^1 + 9 \times 10^0 + 4 \times 10^{-1} + 5 \times 10^{-2}$$

2- النظام الثنائي (Binary System)

- إن الأساس المستعمل في النظام الثنائي هو 2 ويتكون هذا النظام من رقمين فقط هما 0 و 1 ويسمى كل منهما رقماً ثنائياً (Binary Digit).
 - كما يمكن استخدام أي حالتين مثل "صح" و "خطأ" أو "تشغيل" و "إطفاء".
 - لتمثيل كل من الرقمين 0 و 1 يلزم خانة واحدة فقط، ولهذا السبب أصبح من الشائع إطلاق اسم بت (Bit) على الخانة التي يحتلها الرقم داخل العدد الثنائي.
- العدد الثنائي 10

1	0		الأرقام
2^1	2^0	=	أوزان المواقع
2	1	=	قيم الأوزان
	$(2 \times 1) + (1 \times 0)$	=	قيمة العدد
2	+ 0	=	
	2	=	

1-2- التحويل من النظام الثنائي إلى النظام العشري

- لتحويل أي عدد ثنائي إلى مكافئه العشري فإنه يجب علينا استعمال قانون التمثيل الموضعي للأعداد. و ينطبق هذا القانون عندما يكون الرقم الثنائي صحيحاً أو كسراً مع مراعاة أن أساس نظام العد هنا هو 2.

$$N = a_n R^n + a_{n-1} R^{n-1} + \dots + a_1 R^1 + a_0 R^0 + a_{-1} R^{-1} + \dots + a_{-m} R^{-m}$$

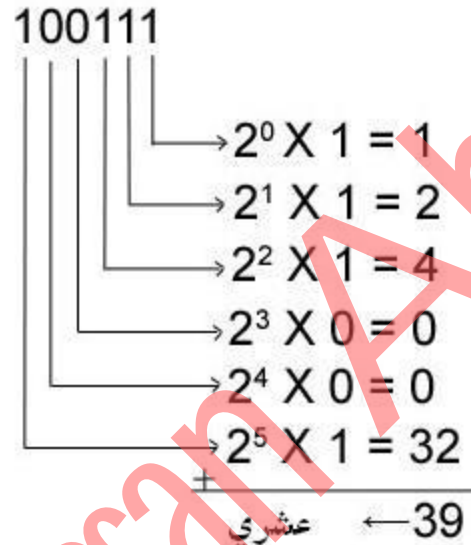
مثال 1: حول العدد الثنائي التالي إلى مكافئه العشري: $(110011010)_2 = (??)_{10}$

$$\begin{aligned} 110011010 &= \\ &= 0 * 2^0 + 1 * 2^1 + 0 * 2^2 + 1 * 2^3 + 1 * 2^4 + \\ &\quad 0 * 2^5 + 0 * 2^6 + 1 * 2^7 + 1 * 2^8 \\ &= 0 + 2 + 0 + 8 + 16 + 0 + 0 + 128 + 256 \\ &= 410 \end{aligned}$$

مثال 2: تحويل العدد الثنائي إلى مكافئه العشري:

تمثيل ثنائي = تمثيل عشري

1	=	0001
2	=	0010
3	=	0011
4	=	0100
5	=	0101
6	=	0110
7	=	0111
8	=	1000
9	=	1001
10	=	1010
11	=	1011
12	=	1100
13	=	1101
14	=	1110
15	=	1111



Examples:

$$10101_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 0 \times 2^1 + 1 \times 2^0 = 16 + 4 + 1 = 21 \text{ (Decimal)}$$

$$10111_2 = 1 \times 2^4 + 0 \times 2^3 + 1 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 16 + 4 + 2 + 1 = 23 \text{ (Decimal)}$$

$$100011_2 = 1 \times 2^5 + 0 \times 2^4 + 0 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = 32 + 2 + 1 = 35 \text{ (Decimal)}$$

2-2- التحويل من النظام العشري إلى النظام الثنائي

تحويل الأعداد العشرية الصحيحة الموجبة :

لتحويل أي عدد صحيح موجب من النظام العشري إلى الثنائي نستعمل طريقة الباقي (Remainder Method) الموضحة كآتي:

1. أقسم العدد العشري على الأساس 2.
2. أحسب باقي القسمة الذي يكون إما 1 أو 0.
3. أقسم ناتج القسمة السابق على الأساس 2 كما في الخطوة (1)
4. أحسب باقي القسمة كما في الخطوة (2)
5. استمر في عملية القسمة وتحديد الباقي حتى يصبح ناتج القسمة الصحيح صفراً.
6. العدد الثنائي المطلوب يتكون من أرقام الباقي مقروءة من الباقي الأخير إلى الباقي الأول (لاحظ أن الباقي الأول يمثل LSD (least significant digit) بينما يمثل الباقي الأخير MSD (most significant digit))

أمثلة: التحويل من النظام العشري إلى النظام الثنائي.

مثال 1: حول الرقم 12 من النظام العشري إلى النظام الثنائي.

	نتاج القسمة	الباقى	
1	$12 \div 2 = 6$	0	الخانة الأدنى منزلة LSD
2	$6 \div 2 = 3$	0	
3	$3 \div 2 = 1$	1	
4	$1 \div 2 = 0$	1	الخانة الأعلى منزلة MSD
		إنهاء القسمة	

فيكون الناتج (من أسفل إلى أعلى ومن اليسار إلى اليمين):

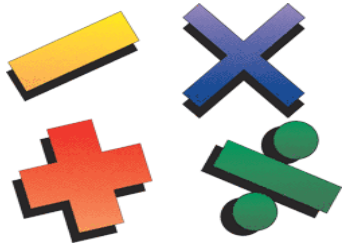
$$(12)_{10} = (1100)_2$$

مثال 2: حول الرقم 150 من النظام العشري إلى النظام الثنائي.

0	2	150
1	2	75
1	2	37
0	2	18
1	2	9
0	2	4
0	2	2
1		1

$$(150)_{10} = (10010110)_2$$

إجراء العمليات الحسابية على الأعداد الثنائية الموجبة



يمكن إجراء العمليات الحسابية من جمع وطرح وضرب وقسمة كما هو الحال في النظام العشري مع مراعاة أن أساس النظام المستعمل هنا هو 2.

عملية الجمع :

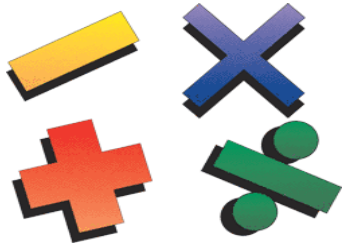


لو أخذنا عددين ثنائيين A, B وكان كل منهما يتكون من خانة واحدة فقط (Bit)، وبما أن كل خانة يمكن أن تكون إما 0 أو 1 فإنه يوجد للعددين معاً أربع احتمالات كالآتي:

A	B	المجموع	المحمول Carry
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

أما إذا كانت الأعداد الثنائية مكونة من أكثر من خانة واحدة فإن عملية الجمع تنفذ بنفس طريقة الجمع

في النظام العشري مع مراعاة أن أساس النظام العد المستعمل هو 2



عملية الجمع +

$$\begin{array}{r}
 \text{المجموع} \\
 \text{العدد الأول} \\
 + \\
 \text{العدد الثاني} \\
 \hline
 111 \\
 101 \\
 + \\
 011 \\
 \hline
 1000
 \end{array}$$

مثال 1: اجمع العددين الثنائيين: $(101)_2 + (011)_2 = (?)_2$

الناتج: $(101)_2 + (011)_2 = (1000)_2$



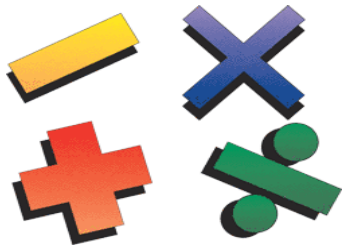
مثال 2: اجمع العددين الثنائيين: $(11)_2 + (11)_2 = (?)_2$

الناتج: $(11)_2 + (11)_2 = (110)_2$

مثال 3: اجمع العددين الثنائيين: $(101101)_2 + (1011)_2 = (?)_2$

$$\begin{array}{r}
 1 \\
 101101 \\
 + \\
 001011 \\
 \hline
 111000
 \end{array}$$

الناتج: $(101101)_2 + (1011)_2 = (111000)_2$



عملية الضرب:

تشبه عملية الضرب في النظام الثنائي عملية الضرب في النظام العشري

$$0 = 0 \times 0$$

$$0 = 1 \times 0$$

$$0 = 0 \times 1$$

$$1 = 1 \times 1$$

مع مراعاة القواعد الأساسية التالية:

مثال 1: ما هو ناتج ضرب العددين الثنائيين:

$$(101)_2 \times (10)_2 = (?)_2$$

$$(101)_2 \times (10)_2 = (1010)_2$$

الناتج هو:

مثال 2: ما هو ناتج ضرب العددين الثنائيين:

$$(101)_2 \times (111)_2 = (?)_2$$

$$(101)_2 \times (111)_2 = (100011)_2$$

الناتج هو:

$$\begin{array}{r} 101 \\ \times 111 \\ \hline 101 \\ 101 \\ 101 \\ + 101 \\ \hline 100011 \end{array}$$

3- النظام الثماني (Octal System)

- نظام الأعداد الثماني هو نظام الأعداد الذي أساسه العدد (8).
- مكونات هذا النظام ستكون: (0، 1، 2، 3، 4، 5، 6، 7).
- لا يستخدم نظام الأعداد الثماني بكثرة في أيامنا هذه، واقتصرت استخداماته على البدايات الأولى لظهور الحواسيب.

• **مثال:** العدد 1753_8 هو: $1 \times 8^3 + 7 \times 8^2 + 5 \times 8^1 + 3 \times 8^0 = (1003)_{10}$

- يستخدم نظام الثماني في الحوسبة في بعض الأحيان كما في ترميز UTF-8 (نظام ترميز Unicode)

- العلاقة بين كل من نظام الأعداد الثماني ونظام الأعداد الثنائي:

7	6	5	4	3	2	1	0
111	110	101	100	011	010	001	000

4- نظام الأعداد الستة عشري (Hexadecimal Number System)

Bit pattern	Hexadecimal representation
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
1010	A
1011	B
1100	C
1101	D
1110	E
1111	F

- يمتلك نظام الأعداد الستة عشري أساساً هو العدد (16).
- مكونات هذا النظام هي الأعداد التالية: (0، 1، 2، 3، 4، 5، 6، 7، 8، 9، A، B، C، D، E، F).
- بهدف التمييز بينه وبين النظام العشري، تم ترميز الأرقام من 10 وحتى 15 في النظام الستة عشري بالأحرف التالية: (A، B، C، D، E، F) حيث تشير هذه الأحرف للأرقام (10، 11، 12، 13، 14، 15) على التوالي.

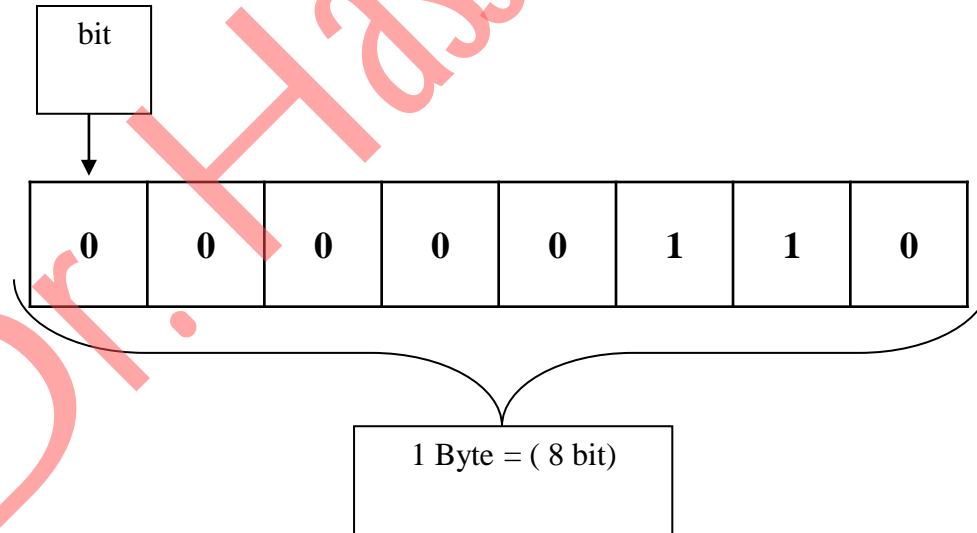
- يستخدم نظام الستة عشري في ترميز عناوين ذاكرة الحاسوب.

مثال: الرقم التالي : 3AD1 يكتب كالتالي:

$$3 \times 16^3 + 10 \times 16^2 + 13 \times 16^1 + 1 \times 16^0 = (14897)_{10}$$

5- تمثيل البيانات داخل الحاسوب.

- **بت (Bit):** أصغر وحدة تخزين في الحاسوب بداخل الذاكرة هي البت (Bit) وهي خلية ثنائية تستوعب فقط اما 0 أو 1 ويعني 0 off و 1 on وهناك مضاعفات لهذه الوحدة.
- **بايت (Byte):** وهي تساوي 8 بت ويستخدم البت لتمثيل رقم أو حرف أو رمز.
- **الكلمة Word:** وهي عبارة عن مجموعة من البتات المتجاورة تعامل كوحدة واحدة ويعتمد عدد البتات في الكلمة على الحاسوب وهو عنصر مهم في تصميم الحاسوب وأصغر كلمة تساوي 8 بت وهناك حواسيب بأطوال كلمة تساوي 16 بت و 32 بت و 64 و 128 بت.



5-1- المتمم الثنائي (Two's Complement) و كيفية الحصول عليه

- يستخدم المتمم الثنائي من أجل تمثيل الأعداد السالبة في الحاسب في النظام الثنائي.
- لتمثيل عدد سالب نتبع الخطوات التالية:

- نكتب العدد بالنظام الثنائي.
- نقلب الأصفار واحداً و الواحدات أصفاراً.
- نضيف واحد (1) إلى الرقم الناتج.

المتمم الأحادي

المتمم الثنائي = المتمم الأحادي + 1

مثال: مثل العدد (-30) في النظام الثنائي عن طريق المتمم الثنائي:

الخطوة الأولى: نوجد المكافئ الثنائي للعدد الصحيح: $(30)_{10} = (00011110)_2$

الخطوة الثانية: نقلب كل 0 إلى 1 وكل 1 إلى 0: $1110\ 0001$

الخطوة الثالثة: نضيف (نجمع) 1 إلى العدد الثنائي الناتج في الخطوة الثانية، أي:

$$\left. \begin{array}{r} 1110\ 0001 \\ + \\ 0000\ 0001 \end{array} \right\} = 1110\ 0010$$

2-5- تمثيل الأعداد داخل الحاسب باستخدام التشفير الثنائي الصريح.

التشفير الثنائي الصريح يتطلب تخزين الأعداد في ذاكرة الحاسب كعدد ثابت من البتات بصيغة **الكلمة**.

- **الكلمة (word)** هي سلسلة من البتات تعامل كوحدة.
- طول الكلمة هو عدد البتات في الكلمة.
- نعتبر أن الحاسوب يستخدم كلمات ذات طول 32 بت.

تمثيل الأعداد الصحيحة

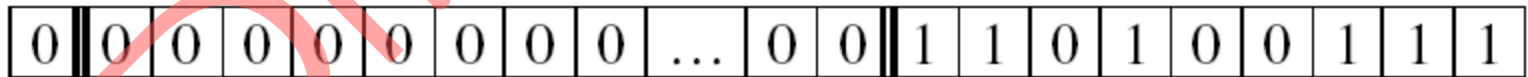
يمثل العدد الصحيح الموجب في ذاكرة الحاسب بصيغته الثنائية، وبمكمل الثنائيات إذا كان سالباً.

مثال: يمثل العدد 423 في ذاكرة حاسب من 32 بت كآتي:

$$\checkmark \text{ نوجد المكافئ الثنائي للعدد المذكور: } (423)_{10} = (11010\ 0111)_2$$

خانة الإشارة

✓ نكتب العدد الثنائي الحاصل بصيغة 32 بت كآتي:



تابع المثال: تمثيل العدد (-423) في ذاكرة حاسب من 32 بت كالاتي:

✓ نوجد المكافئ الثنائي للعدد الموجب 423: $(423)_{10} = (110100111)_2$

✓ نكتب العدد الحاصل بصيغة 32 بت: $[00\dots(110100111)]_{32}$

✓ نقلب كل 0 إلى 1 وكل 1 إلى 0. $[00\dots(110100111)]_{32} \rightarrow [11\dots(001011000)]_{32}$

✓ نضيف (نجمع) 1 إلى العدد الثنائي الناتج: $[11\dots(001011000)]_{32} + 1 = [11\dots(001011001)]_{32}$

خانة الإشارة

✓ الناتج بصيغة 32 بت:

1	1	1	1	1	1	1	1	1	1	1	...	1	1	0	0	1	0	1	1	0	0	1
---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	---	---	---	---	---	---

• يستطيع الحاسب معرفة هل الرقم في الذاكرة موجب أم سالب وذلك بالنظر إلى البت في الخانة الأولى من أقصى اليسار: إذا كان البت الأول 0 فإن العدد موجب وإذا كان 1 فإن العدد سالب.

• بالتالي، فإن أكبر عدد صحيح (موجب) يمكن تخزينه في ذاكرة الحاسب ذي 32 بت هو:

0	1	1	1	1	1	1	1	1	1	1	1	...	1	1	1	1	1	1	1	1	1	1
---	---	---	---	---	---	---	---	---	---	---	---	-----	---	---	---	---	---	---	---	---	---	---

• أي 31 واحد، أو $2^{31} - 1$ ويساوي تقريباً 2 مليار عدد موجب.

• بالمثل فإن أصغر عدد (سالب) يمكن تخزينه في ذاكرة الحاسب ذي 32 بت هو: تقريباً 2 مليار عدد سالب.

6- شيفرة الحاسب الآلي (Computer codes)

- يتشكل كل رمز أو حرف أو رقم أو إشارة من 8 بتات (8 Bit = 1 Byte)، وتكون إما 0 أو 1.
- أو البايت (Byte) هو عبارة عن حرف أو رمز أو إشارة ويساوي 8 بت (أي مجموعة من ثمان بتات). ويعتبر البايت وحدة خزن للبيانات في الحاسوب.
- قام المبرمجون بوضع ترميز قياسي (Standard Code) لترميز الحروف الأبجدية والأرقام والرموز فيما يماثله من هذا النظام الثنائي، وعليه فقد استخدم علم الترميز (التشفير) في وضع شفرة لأبجديات اللغة المستعملة (اللغة الإنجليزية مثلاً) وقد تعددت هذه الشيفرات.
- مثلاً حرف a يشفر بنظام أسكي (01100001) ASCII وعلامة (=) (00111101) وهكذا.

1. نظام بي سي دي BCD (Binary Coded Decimal) الذي وضعته شركة IBM ولكن لم يدم طويلاً وذلك نظراً لإقتصار هذا النظام على سداسية البتات والذي يؤدي إلى عدم الترميز (التشفير) لكل الحروف.

2. نظام تشفير إبيديك EBCDIC (Extended Binary-Coded Decimal Interchange Code) و عبارة عن 8 بتات (4 المنطقة و 4 الرقم)، ويكون مسبق بالرمز 11 في الأحرف أو 111 أو 1111 للأرقام، ويستطيع ترميز 256 حرف.

3. نظام تشفير أسكي ASCII (American Standard Code for Information Interchange) وعملت هذه التشفيرة في الوقت الذي تطورت فيه الحواسيب الصغيرة والذي مكن مؤسسة المعايير القياسية الأمريكية (American National Standards Institute) من تحديد المعايير القياسية للحواسيب. وهو نظام سباعي الخانات أضيف له الصفر في الخانة الثامنة اليسرى.

4. تعتبر مجموعة ترميز ASCII حالياً أكثر المجموعات شهرة ، وقد تم ترميز أبجديات اللغة في الشيفرتين السابقتين استخدمتا ثمان خانوات لتشفير الأبجديات والأرقام ، فمثلاً رمز الحرف M بتشفير

أسكي هو 01101101 وهكذا.

5. الترميز العالمي Unicode ويسمى المعيار القياسي العالمي للحروف (Unicode Worldwide Character Standard) .

- هو إصدار قياسي جديد يمثل حجم الكلمة ببايتين (Bytes 2) بدلاً من بايت واحد، أي أن 16 بت لكل حرف ولذا فإن احتمال وجود الحرف من بين $2^{16} = 65536$ حرفاً أو رمزاً حقيقياً.
- هذا يكفي لتغطي كافة لغات العالم أي أنها تمثل أي رمز في العالم ويمكن ضم الحروف الهجائية الصينية واليابانية والكورية وتلك الموجودة في النصوص الكلاسيكية والتاريخية المعروفة، بما فيها العربية .

ASCII	SYMBOL	EBCDIC	ASCII	SYMBOL	EBCDIC
00110000	0	11110000	01001110	N	11010101
00110001	1	11110001	01001111	O	11010110
00110010	2	11110010	01010000	P	11010111
00110011	3	11110011	01010001	Q	11011000
00110100	4	11110100	01010010	R	11011001
00110101	5	11110101	01010011	S	11100010
00110110	6	11110110	01010100	T	11100011
00110111	7	11110111	01010101	U	11100100
00111000	8	11111000	01010110	V	11100101
00111001	9	11111001	01010111	W	11100110
01000001	A	11000001	01011000	X	11100111
01000010	B	11000010	01011001	Y	11101000
01000011	C	11000011	01011010	Z	11101001
01000100	D	11000100	00100001	!	01011010
01000101	E	11000101	00100010	"	01111111
01000110	F	11000110	00100011	#	01111011
01000111	G	11000111	00100100	\$	01011011
01001000	H	11001000	00100101	%	01101100
01001001	I	11001001	00100110	&	01010000
01001010	J	11010001	00101000	(01001101
01001011	K	11010010	00101001)	01011101
01001100	L	11010011	00101010	*	01011100
01001101	M	11010100	00101011	+	01001110



نهاية المحاضرة السابعة