



פרק 2

שיטות ספירה וייצוג מידע במחשב

► בפרק זה נלמד:

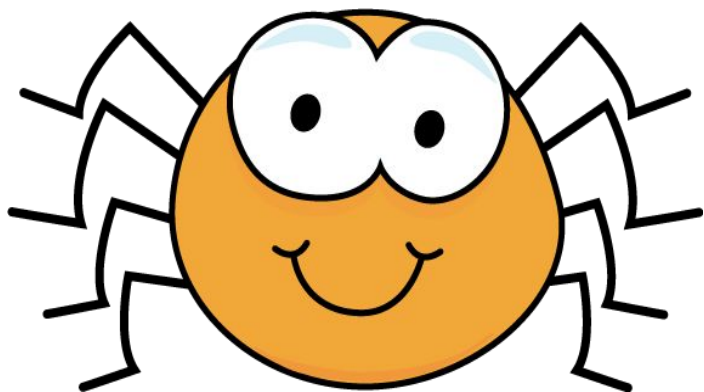
- שיטות ספירה, בדגש על בסיס 2 ו-16
- המרות בין בסיסים
- פעולות חיבור, חיסור, כפל וחילוק בבסיסים שונים
- שיטת המשלים ל-2 לייצוג מספרים Signed ו-Uncolored
- גדלים בזיכרון המחשב: ביט, Nibble, בית...
- קוד ASCII לייצוג תווים ע"י 8 ביט

- ▶ לבני אדם יש עשר אצבעות
- ▶ שיטת הספירה הדצימלית – בסיס 10 – מבוססת על עשר ספרות
 - 0,1,2,3,4,5,6,7,8,9
 - בשביל לייצג מספרים הגדולים מ-9 נדרשות שתי ספרות
- ▶ אפשר לספור גם בבסיסים אחרים!
- ▶ למה זה חשוב? בגלל האופן שבו מידע מיוצג במחשב

ספירה בבסיסים שונים

בסיס 3 0,1,2	בסיס 8 0,1,2,3,4,5,6,7	בסיס 10 0,1,2,3,4,5,6,7,8,9
0	0	0
1	1	1
2	2	2
10	3	3
11	4	4
12	5	5
20	6	6
21	7	7
22	10	8
100	11	9
101	12	10
102	13	11

כמה רגליים יש לעכביש?



- ▶ בבסיס עשר – 8 רגליים
- ▶ בבסיס שמונה – 10 רגליים
- ▶ בבסיס שלוש – 22 רגליים

▶ כמות הרגליים לא השתנתה, רק הייצוג שלה.

משחק – pearls before swine III

► www.transience.com.au/pearl3.html

- נקדיש כמה דקות לחוקי המשחק
- מאוחר יותר נבין את הקשר לשיטות ספירה



רישום בסיס הספירה

► מעכשיו נציין באיזה בסיס ספירה מדובר

501_{10}

► זאת כדי למנוע טעויות כגון

10_{10} ----- 10_2

47_{10} ----- 47_8

מיקום לפי ערך

▶ הערך של ספרה נקבע לפי המיקום שלה

◦ 501 לעומת 105

▶ בבסיס עשר, ערך המיקום נקבע לפי חזקות של עשר

▶ כמה דוגמאות בבסיס עשר:

$$47_{10} = 7 \cdot 10^0 + 4 \cdot 10^1$$

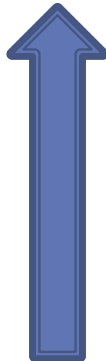
$$375_{10} = 5 \cdot 10^0 + 7 \cdot 10^1 + 3 \cdot 10^2$$

$$1994_{10} = 4 \cdot 10^0 + 9 \cdot 10^1 + 9 \cdot 10^2 + 1 \cdot 10^3$$

המרת מספר עשרוני לבסיס אחר

► מה ערכו של המספר 199 בבסיס 5?

שארית	פעולה
4	$39 = 199:5$
4	$7 = 39:5$
2	$1 = 7:5$
1	$0 = 1:5$



► קוראים את השאריות מלמטה למעלה

► $199_{10} = 1244_5$

השיטה הבינארית – בסיס 2

- ▶ קיימות שתי ספרות בלבד – 0,1
 - המספר 2 צריך להיות מיוצג ע"י יותר מספרה אחת
- ▶ ערך המיקום של ספרה נקבע לפי חזקות של 2

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1

▶ לדוגמה המספר $19_{10} = 16 + 2 + 1 = 10011_2$

2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
128	64	32	16	8	4	2	1
-	-	-	1	0	0	1	1

המרה מעשרוני לבינארי

► נבצע את ההמרה ההפוכה, מ- 19_{10} לבינארי

שארית	פעולה
1	$9 = 19:2$
1	$4 = 9:2$
0	$2 = 4:2$
0	$1 = 2:2$
1	$0 = 1:2$



► קיבלנו 10011_2 😊

השיטה ההקסדצימלית – בסיס 16

- קיימות 16 ספרות
- לוקחים שש אותיות ונותנים להן ערך מספרי:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F	E	D	C	B	A	9	8	7	6	5	4	3	2	1	0

- מספר יכול להיות צירוף של אותיות וספרות:

$F16_{16}$

$C1A_{16}$

$CODE_{16}$

$COFFEE_{16}$

שיטות רישום מספרים הקס'

- כתיבת הבסיס למטה: $CODE_{16}$
- ▶ תוספת 0 לפני המספר: $0xCODE$
- ▶ סיומת h בסוף המספר
- אם המספר מתחיל באות, מוסיפים 0 בתחילת המספר
- $0CODEh$

המרה בין הקסדצימלי לדצימלי

► מהקסדצימלי לדצימלי:

► $4F_{16} = F \cdot 16^0 + 4 \cdot 16^1 = 15 + 64 = 79_{10}$

► מדצימלי להקסדצימלי:

שארית	פעולה
7	$12 = 199 : 16$
12(C)	$0 = 12 : 16$



► ולכן $C7_{16} = 199_{10}$

המרה בין בינארי והקסדצימלי

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

▶ להמרה בין בינארי
והקסדצימלי יש תכונה
מיוחדת: כל ספרה הקס' היא
ארבע ספרות בינאריות

המרה בין בינארי והקס' – דוג'

$$1001_2 = 9_{16} \quad \blacktriangleright$$

$$B_{16} = 1011_2 \quad \blacktriangleright$$

$$9B_{16} = 10011011_2 \quad \blacktriangleright$$

▶ מעבר בין בינארי להקס' הוא פשוט מאוד

▶ כתיב בהקס' פשוט יותר לקריאה ולזכירה מאשר
בינארי

▶ לכן הקס' שימושי בעבודה עם מחשבים

תרגילים – מעבר בין בסיסים

בסיס 10	בסיס 2	בסיס 16
35		
	1011	
		12
	1100011	
59		
		63
		5C
	11110	
21		
		31

$$\begin{array}{r} 1 \\ + 133 \\ \underline{70} \\ 203 \end{array}$$

▶ בחיבור בבסיס 10, מעבירים נשא
(Carry) כשהתוצאה גדולה מ-9

$$\begin{array}{r} 1 \\ + 1010 \\ \underline{11} \\ 1101 \end{array}$$

▶ בחיבור בבסיס 2, מעבירים נשא
כשהתוצאה גדולה מ-1

$$\begin{array}{r} 1 \\ + ABCD \\ \underline{0123} \\ ACF0 \end{array}$$

▶ בחיבור בבסיס 16, מעבירים נשא
כשהתוצאה גדולה מ-15 (0Fh)

תרגילים – חיבור

A	B	A+B
25	F	
B	29	
26	2C	
34	34	
1F	2A	
1E	12	
22	2F	

A	B	A+B
100111	10100	
11001	11000	
110000	100101	
111001	11001	
110110	101011	
111001	10011	
111001	101010	

- ▶ אם הספרה העליונה גדולה מהתחתונה – חיסור פשוט
- ▶ אם הספרה העליונה שווה לתחתונה – התוצאה אפס
- ▶ אם הספרה העליונה קטנה מהתחתונה – "נשא שלילי"

$$\begin{array}{r}
 \text{10} \text{ } ^{-1} \\
 619 \\
 - \quad 21 \\
 \hline
 598
 \end{array}$$

חיסור- המשך

• בבסיס 16:

$$\begin{array}{r} -1 \ 16 \\ -1 \ 16 \\ - \text{DEAD} \\ \underline{\text{CODE}} \\ 1\text{DCF} \end{array}$$

► בבסיס 2:

$$\begin{array}{r} -1 \ 10 \\ \underline{1010} \\ - \\ \underline{1} \\ 1001 \end{array}$$

תרגילים – חיסור

A	B	A-B
101010	10100	
11001	10110	
10100	1101	
100101	10011	
111000	1111	
101011	11010	
10101	1010	

A	B	A-B
93	3D	
130	22	
E7	60	
C3	19	
CF	56	
47	12	
54	D	

▶ בבסיס 10:

- לוח הכפל בגודל 10×10
- כשכופלים בעשר, מזיזים את כל הספרות שמאלה

▶ בבסיס 2:

- לוח הכפל בגודל 2×2
- כשכופלים בשתיים, מזיזים את כל הספרות שמאלה

X	0	1
0	0	0
1	0	1

► כפל בבסיס 2 (10 כפול 3):

$$\begin{array}{r} 1010 \\ \times 11 \\ \hline 1010 \\ 1010- \\ \hline 11110 \end{array}$$

► כפל בבסיס 16: מעבירים לבסיס 2 וממשיכים

A	B	AxB
111	1011	
1001	1001	
100	1010	
100	111	
1110	1100	
1001	11	
111	1110	

חוקי החילוק בבסיס 2: $\frac{0}{1} = 0$ $\frac{1}{1} = 1$

תרגיל - $10110_2 / 101_2$ ($22_{10} / 5_{10}$):

$$\begin{array}{r} 100 \\ \overline{10110} \overline{) 101} \\ - 101 \\ \hline 010 \end{array}$$

תרגילים – חילוק

A	B	A/B	שארית
100000	110		
10101	110		
100110	1001		
10101	100		
101010	10		
111001	110		
101000	1001		

ייצוג מספרים במחשב

N	המספר הכי גדול שאפשר לייצג	תרגום לעשרוני
1	1	1
2	11	3
3	111	7
4	1111	15
5	11111	31
6	111111	63
7	1111111	127
8	11111111	255

▶ כל ספרה בינארית נקראת

Bit: **B**inary **D**igit

▶ בזיכרון המחשב יש תאים

שבהם כמות קבועה של
ביטים

▶ תא זיכרון בגודל N ביטים

יכול לייצג מספר עד $2^N - 1$

ייצוג מספרים במחשב – המשך

- ▶ מה יקרה אם ננסה להכניס ל-N ביטים מספר יותר גדול ממה שניתן לייצג ע"י N ביטים?
- ▶ דוגמה: $255+1$

$$\begin{array}{r}
 + 11111111 \\
 \underline{00000001} \\
 (1)00000000
 \end{array}$$

- את התוצאה אי אפשר לשמור ע"י 8 ביטים
- הזיכרון יכול 00000000
- במקום אחר בזיכרון יודלק ביט שאומר שהיה נשא בפעולה האחרונה

ייצוג מספרים עם סימן – signed

- ▶ עד כה חשבנו רק על מספרים חסרי סימן – Unsigned
- ▶ הצלחנו לייצג מספרים חיוביים בלבד
 - $1 = 00000001$
 - $2 = 00000010$
 - וכו'
- ▶ שיטות לייצוג מספרים מסומנים – Signed:
 - גודל וסימן
 - משלים ל-1
 - משלים ל-2

שיטת גודל וסימן

▶ העקרון: הביט השמאלי מייצג את הסימן

◦ 0- המספר חיובי

◦ 1- המספר שלילי

▶ יתר הביטים מייצגים את הגודל

▶ לדוגמה **00000011**

◦ סימן 0- חיובי

◦ גודל **00000011** – 3

▶ ייצוג של מינוס 3:

◦ **10000011**

שיטת גודל וסימן – המשך

מספר עשרוני	ייצוג בשיטת גודל וסימן	מספר עשרוני	ייצוג בשיטת גודל וסימן
7	0111	-7	1111
6	0110	6-	1110
5	0101	5-	1101
4	0100	4-	1100
3	0011	3-	1011
2	0010	2-	1010
1	0001	1-	1001
0	0000	0-	1000

שיטת גודל וסימן – המשך

- ▶ יתרון השיטה – קל להמיר בין מספרים חיוביים ושליילים
- ▶ חסרון השיטה – חיבור של מספר עם הנגדי שלו אינו מסתכם באפס:

$$\begin{array}{r} 0011 \\ + 1011 \\ \hline 1110 \end{array}$$

שיטת המשלים ל-1

► הנגדי של מספר הוא היפוך הביטים שלו:

◦ 0 הופך ל-1

◦ 1 הופך ל-0

מספר עשרוני	ייצוג בשיטת המשלים ל-1	מספר עשרוני	ייצוג בשיטת המשלים ל-1
7	0111	-7	1000
6	0110	6-	1001
5	0101	5-	1010
4	0100	4-	1011
3	0011	3-	1100
2	0010	2-	1101
1	0001	1-	1110
0	0000	0-	1111

שיטת המשלים ל-1 – המשך

► כלל החיבור: אם יש נשא – מוסיפים אותו לביט הימני

$$\begin{array}{r}
 \begin{array}{cc} 1 & 1 \end{array} \\
 0101 \\
 + \\
 \underline{1101} \\
 (1)0010 \\
 0011
 \end{array}$$

שיטת המשלים ל-1 – המשך

▶ יתרונות השיטה:

- קל למצוא את הנגדי של מספר
- פעולות החשבון יוצאות נכון

▶ חסרונות השיטה:

- כדי לבצע הוספה של הנשא לביט הימני צריך חישוב נוסף
- יש לאפס שני ייצוגים – 1111111,00000000

שיטת המשלים ל-2

▶ הנגדי של מספר הוא היפוך הביטים שלו,
פלוס 1

▶ לדוגמה, נמצא את הנגדי של שש (00000110):

$$\begin{array}{r}
 + \quad 11111001 \\
 \quad \quad \quad 1 \\
 \hline
 11111010
 \end{array}$$

▶ שש ועוד מינוס שש:

$$\begin{array}{r}
 + \quad 00000110 \\
 \quad 11111010 \\
 \hline
 (1)00000000
 \end{array}$$

שיטת המשלים ל-2 – המשך

מספר עשרוני	ייצוג בשיטת המשלים ל-2	מספר עשרוני	ייצוג בשיטת המשלים ל-2
7	0000 0111	-1	1111 1111
6	0110 0000	2-	1110 1111
5	0101 0000	3-	1101 1111
4	0100 0000	4-	1100 1111
3	0011 0000	5-	1011 1111
2	0010 0000	6-	1010 1111
1	0001 0000	7-	1001 1111
0	0000 0000	8-	1000 1111

► בעזרת N ביטים, ניתן לייצג את התחום שבין $2^{N-1}-1$ לבין 2^{N-1}

◦ 8 ביטים: $128- < 127+$

◦ 16 ביטים: $32,768- < 32,767+$

◦ וכו'

שיטת המשלים ל-2 – המשך

▶ חסרונות:

- הנגדי של מספר אינו ברור כמו בשיטות האחרות

▶ יתרונות:

- פעולות החשבון נותנות תוצאות הגיוניות
- ייצוג יחיד לאפס

▶ המספרים במחשב מיוצגים בשיטת המשלים ל-2

המרת מספר signed מבינארי לעשרוני

- ▶ אם המספר חיובי (ביט שמאלי 0):
 - המרה "רגילה"
 - לכל ביט יש מיקום
 - הערך נקבע לפי 2 בחזקת המיקום
- ▶ אם המספר שלילי (ביט שמאלי 1):
 - מוצאים את הנגדי של המספר
 - מוסיפים מינוס

המרת מספר signed לבסיס עשר – דוגמה

► מה מייצג המספר 10111111?

- הביט השמאלי הוא 1, לכן מדובר במספר שלילי
- נמצא את הנגדי שלו ונוסיף לו סימן מינוס

מספר מקורי: 10111111

משלים ל-1: 01000000

משלים ל-2: 01000001

$$65 = 2^0 + 2^6$$

לכן המספר מייצג -65

ייצוג signed לעומת ייצוג unsigned

- ▶ ראינו שהמספר 1011111 שווה -65
- ▶ למה שווה המספר 1011111 בייצוג unsigned?
 - $128+32+16+8+4+2+1 = 191$
- ▶ אז למה שווה 1011111?
 - תלוי בפרשנות שלנו! אפשר לפרש את הספרות הבינאריות כך או כך
 - פרשנות כ-65: signed
 - פרשנות כ-191: unsigned

ייצוג מידע במחשב

- ▶ סיבית - Bit
- ▶ רביעיית ביטים - Nibble
- ▶ בית (8 ביטים) - Byte
- ▶ מילה (16 ביטים) - Word
- ▶ מילה כפולה (32 ביטים) - Double Word

▶ ביט שומר אחד משני ערכים:

0 ◦

1 ◦

▶ ביט יכול לייצג כל שני ערכים שונים זה מזה:

◦ 0 מייצג "שקר", 1 מייצג "אמת"

◦ 0 "ירוק", 1 "אדום"

◦ 0 "517", 1 "348"

▶ המשמעות תלויה בפרשנות שלנו

רביעיית ביטים Nibble

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

- יש 16 Nibble שונים
- לכל Nibble מתאימה ספרה הקס'
- לכן Nibble שימושי לקריאה של רצפי ביטים

1101 1110 1010 1101 1100 0000 1101 1110
D E A D C 0 D E

- ▶ 8 ביטים
- ▶ היחידה הקטנה ביותר שיש לה כתובת בזיכרון
- ▶ סידור הביטים:

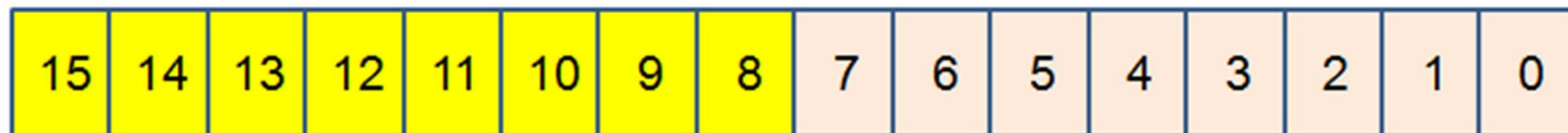


- ▶ ביט מספר 0 הוא Low Order Bit
- ▶ ביט מספר 7 הוא High Order Bit

▶ 16 ביטים

▶ 2 בתים

▶ ניתן לייצג 2 בחזקת 16 = 65,536 ערכים שונים



High Order Byte

Low Order Byte

מילה כפולה Double Word

- ▶ 32 ביטים
- ▶ 4 בתים
- ▶ שתי מילים
- ▶ 2 בחזקת 32 = 4,294,967,296 ערכים שונים

American Standard Code for Information Interchange

קוד נפוץ לייצוג תווים

כל 8 ביט מייצגים תו – סה"כ 255 תווים

Regular ASCII Chart (character codes 0 - 127)															
000	<nul>	016	<dle>	032	sp	048	0	064	@	080	P	096	`	112	p
001	<soh>	017	<dc1>	033	!	049	1	065	A	081	Q	097	a	113	q
002	<stx>	018	<dc2>	034	"	050	2	066	B	082	R	098	b	114	r
003	<etx>	019	<dc3>	035	#	051	3	067	C	083	S	099	c	115	s
004	<eot>	020	<dc4>	036	\$	052	4	068	D	084	T	100	d	116	t
005	<enq>	021	<nak>	037	%	053	5	069	E	085	U	101	e	117	u
006	<ack>	022	<syn>	038	&	054	6	070	F	086	V	102	f	118	v
007	<bel>	023	<etb>	039	'	055	7	071	G	087	W	103	g	119	w
008	<bs>	024	<can>	040	<	056	8	072	H	088	X	104	h	120	x
009	<tab>	025		041	>	057	9	073	I	089	Y	105	i	121	y
010	<lf>	026	<eof>	042	*	058	:	074	J	090	Z	106	j	122	z
011	<vt>	027	<esc>	043	+	059	;	075	K	091	[	107	k	123	{
012	<np>	028	<fs>	044	,	060	<	076	L	092	\	108	l	124	
013	<cr>	029	<gs>	045	-	061	=	077	M	093	]	109	m	125	}
014	<so>	030	<rs>	046	.	062	>	078	N	094	^	110	n	126	~
015	<si>	031	<us>	047	/	063	?	079	O	095	_	111	o	127	Δ

קוד ASCII – דוגמה

► מהו הייצוג של “HELLO WORLD”!

H E L L O W O R L D !

48 45 4C 4C 4F 20 57 4F 52 4C 44 21

```
ds:0000 48 45 4C 4C 4F 20 57 4F HELLO WO
ds:0008 52 4C 44 21 00 00 00 00 RLD!
ds:0010 00 00 00 00 00 00 00 00
ds:0018 00 00 00 00 00 00 00 00
ds:0020 00 00 00 00 00 00 00 00
```

תרגיל – קוד ASCII

► כיתבו את שמכם הפרטי + משפחה באנגלית, בקוד ASCII.

- יש לרשום גם את הקוד של תו הרווח
- אות ראשונה בשם פרטי ובשם משפחה – אות גדולה

סיכום – מה למדנו?

► בפרק זה למדנו–

- שיטות ספירה, בדגש על בסיסים 2 ו-16
- המרות בין בסיסים
- פעולות חיבור, חיסור, כפל וחילוק בבסיסים שונים
- שיטת המשלים ל-2 לייצוג מספרים Signed ו-Unsigned
- גדלים בזיכרון המחשב: ביט, Nibble, בית...
- קוד ASCII לייצוג תווים ע"י 8 ביט