

3.1 Κωδικοποίηση χαρακτήρων

The screenshot shows a window titled "ASCII Encoding" with a toolbar containing navigation arrows, character buttons (A, Z, a, z, 0), an "Info" button, and an "Exit" button. The main area is split into two sections: "Binary" and "ASCII".

Binary: A grid displays the binary sequence 01000001. Below the grid are eight bit controls, each with a "0" and "1" radio button. The current bit settings are: Bit7 (0), Bit6 (0), Bit5 (0), Bit4 (0), Bit3 (0), Bit2 (0), Bit1 (0), and Bit0 (1).

ASCII: A grid displays the ASCII character '1' in green on a blue background.

Text: This program shows the relationship between specific bit sequences and their interpretation. In this case the bit combinations are interpreted as ASCII code (American Standard Code of Information Interchange). Try to set all digits from 0 to 9 by properly adjusting the individual bits.

ΕΙΣΗΓΗΤΗΣ: ΣΤΑΧΟΣ ΚΥΡΙΑΚΟΣ - ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ

Εισαγωγή

Προσπάθησε να μετρήσεις πόσα διαφορετικά γράμματα (κεφαλαία, πεζά), αριθμούς και σύμβολα θα μπορούσες να εισάγεις χρησιμοποιώντας το πληκτρολόγιο του υπολογιστή σου.

Μην ξεχάσεις να υπολογίσεις όλα τα σύμβολα που μοιράζονται κάποια πλήκτρα. Όλα τα γράμματα (α, Δ, κ, Μ, ...), οι αριθμοί (0, 1, 2, 3, ...) τα σύμβολα (#, !, @, ...) είναι χαρακτήρες.

Πόσα δυαδικά ψηφία (bit) χρειάζεσαι για να αναπαραστήσεις όλους τους χαρακτήρες που μέτρησες στο πληκτρολόγιο;

Αν προσθέσεις και τα λατινικά γράμματα (κεφαλαία, πεζά) τι παρατηρείς;

Αναπαράσταση διαφορετικών τιμών με τη χρήση K bits

Με 3 bits μπορούμε να αναπαραστήσουμε 8 διαφορετικές καταστάσεις

000
001
010
011
100
101
111

$$2*2*2 = 8$$

Με 4 bits μπορούμε να αναπαραστήσουμε 16 διαφορετικές καταστάσεις

0000 1000
0001 1001
0010 1010
0011 1011
0100 1100
0101 1101
0111 1111

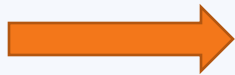
$$2*2*2*2 = 16$$

Γενικά με K bits έχουμε 2^K μοναδικές αναπαραστάσεις

Παραδείγματα

Έστω ότι θέλουμε να αναπαραστήσουμε 240 διαφορετικά μηνύματα με «δυναδικές» λέξεις. Πόσα δυαδικά ψηφία πρέπει να έχει αυτή η λέξη;

$$2^7=128$$



$$2^8=256$$

κάθε λέξη πρέπει να αποτελείται από 8 bits.

Πόσα διαφορετικά μηνύματα μπορούμε να αναπαραστήσουμε με τη χρήση 16 bits;

$$2^{16} = 65536$$



μπορούμε να αναπαραστήσουμε 65536 διαφορετικά μηνύματα

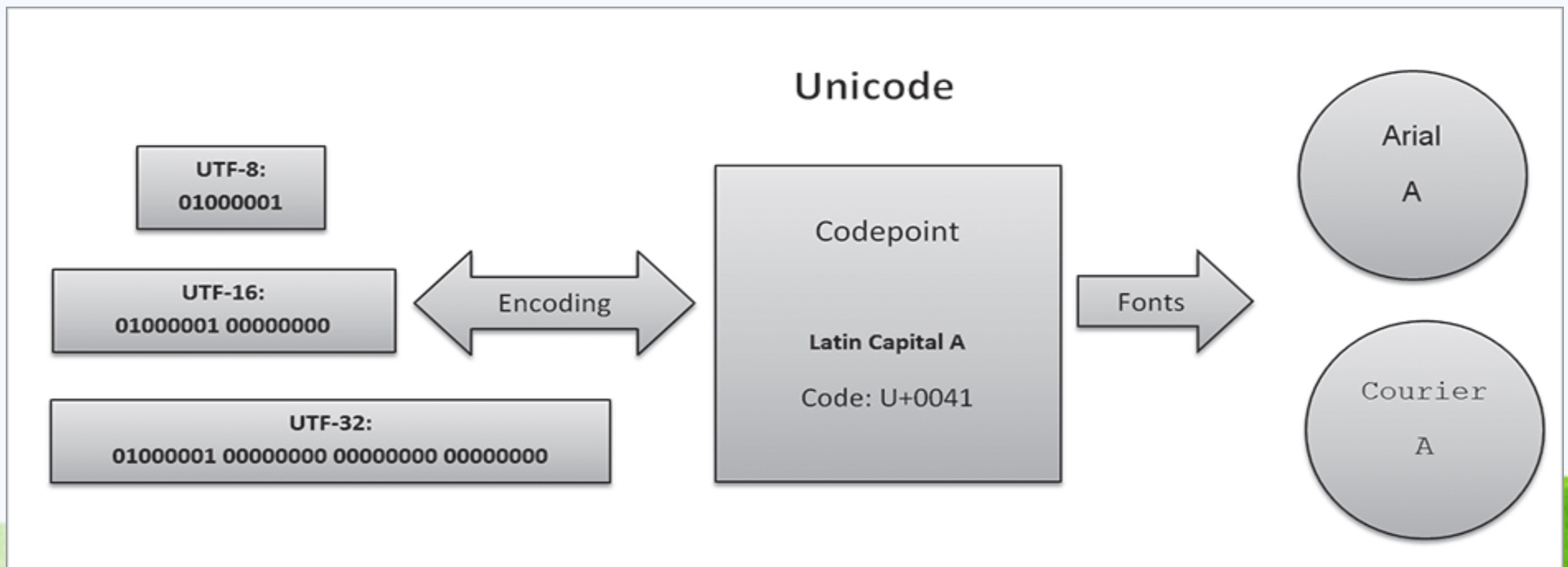
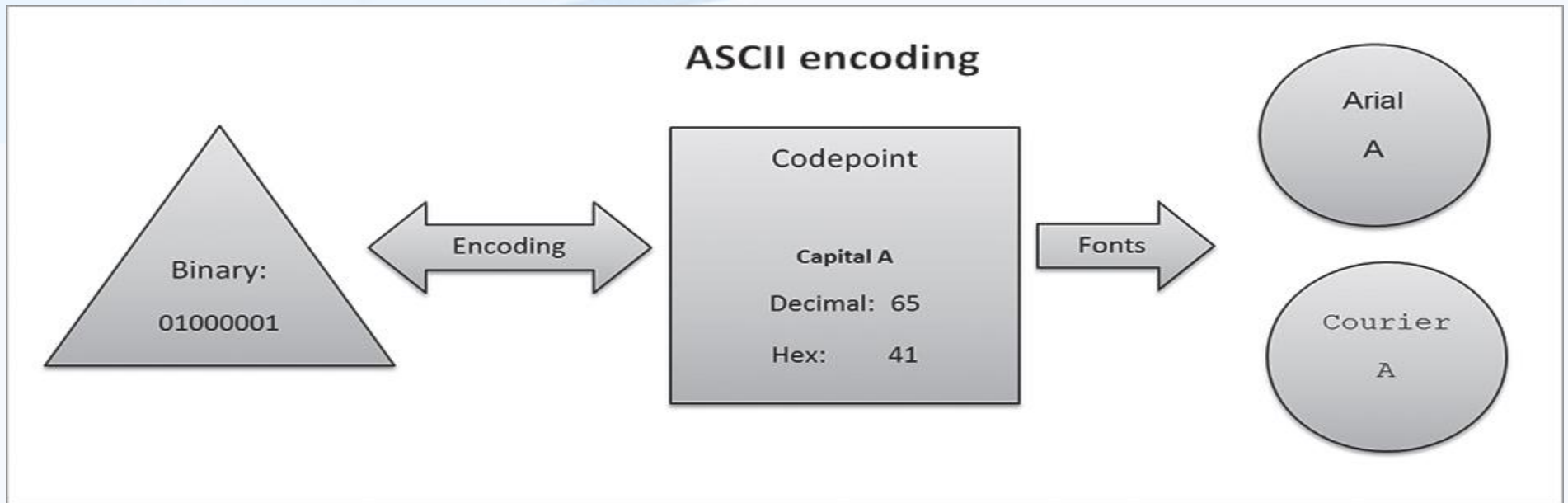
Κωδικοποίηση χαρακτήρων

Οι λέξεις και οι φράσεις σε ένα κείμενο δημιουργούνται από ένα σύνολο χαρακτήρων που περιλαμβάνει αλφαριθμητικά, αριθμούς και άλλα σύμβολα.

Κωδικοποιημένο σύνολο χαρακτήρων ονομάζουμε ένα σύνολο χαρακτήρων όταν κάθε χαρακτήρας αντιστοιχίζεται σε ένα συγκεκριμένο και μοναδικό αριθμό που ονομάζεται **code point**.

Επομένως τα μέσα και οι συσκευές εισόδου, εξόδου, αποθήκευσης και επικοινωνίας μεταφέρουν, αποθηκεύουν και χειρίζονται δεδομένα κειμένου υπό τη μορφή ενός κώδικα χαρακτήρων που **απεικονίζει κάθε χαρακτήρα σε έναν μοναδικό αριθμό** (code point). Η κωδικοποίηση μετατρέπει αυτούς τους αριθμούς σε δυαδική μορφή, ώστε κάθε χαρακτήρας να αντιστοιχίζεται σε μία μοναδική διαφορετική ακολουθία δυαδικών ψηφίων (σειρές από bytes).

Κωδικοποίηση χαρακτήρων



Κώδικας χαρακτήρων ASCII - 1

Ο κώδικας **ASCII** (**American Standard Code for Information Interchange**, Αμερικανικός Πρότυπος Κώδικας για Ανταλλαγή Πληροφοριών) είναι ένα κωδικοποιημένο σύνολο χαρακτήρων του λατινικού αλφάβητου.

Αναπτύχθηκε στη δεκαετία του 1960 από τον Αμερικάνικο Οργανισμό Προτυποποίησης, ώστε να επιτρέπει σε ψηφιακές συσκευές να επικοινωνούν μεταξύ τους και να επεξεργάζονται, να αποθηκεύουν και να μεταδίδουν πληροφορίες σχετικές με χαρακτήρες.

Κώδικας χαρακτήρων ASCII - 2

Αρχικά στον κώδικα ASCII χρησιμοποιούνταν 7 bit για την κωδικοποίηση των χαρακτήρων και το 8ο bit για έλεγχο ορθότητας κατά τη μεταφορά στοιχείων. Το bit αυτό ονομάστηκε **ψηφίο ισοτιμίας (parity bit)**.

Το αποτέλεσμα ήταν να επιτρέπει την απεικόνιση 128 (2⁷) διαφορετικών χαρακτήρων. Σε κάθε χαρακτήρα αντιστοιχίζεται ένας μοναδικός αριθμός από το 0 έως το 127.

0 - 31	χαρακτήρες ελέγχου (μη εκτυπώσιμοι)
32 - 63	αριθμοί, κενό, σημεία στίξης, σύμβολα πράξεων
64 - 95	κεφαλαία λατινικά γράμματα και ειδικά σύμβολα
96 - 127	πεζά λατινικά γράμματα και ειδικά σύμβολα

Κώδικας χαρακτήρων ASCII - 3

Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char	Decimal	Hex	Char
0	0	[NULL]	32	20	[SPACE]	64	40	@	96	60	`
1	1	[START OF HEADING]	33	21	!	65	41	A	97	61	a
2	2	[START OF TEXT]	34	22	"	66	42	B	98	62	b
3	3	[END OF TEXT]	35	23	#	67	43	C	99	63	c
4	4	[END OF TRANSMISSION]	36	24	\$	68	44	D	100	64	d
5	5	[ENQUIRY]	37	25	%	69	45	E	101	65	e
6	6	[ACKNOWLEDGE]	38	26	&	70	46	F	102	66	f
7	7	[BELL]	39	27	'	71	47	G	103	67	g
8	8	[BACKSPACE]	40	28	(72	48	H	104	68	h
9	9	[HORIZONTAL TAB]	41	29)	73	49	I	105	69	i
10	A	[LINE FEED]	42	2A	*	74	4A	J	106	6A	j
11	B	[VERTICAL TAB]	43	2B	+	75	4B	K	107	6B	k
12	C	[FORM FEED]	44	2C	,	76	4C	L	108	6C	l
13	D	[CARRIAGE RETURN]	45	2D	-	77	4D	M	109	6D	m
14	E	[SHIFT OUT]	46	2E	.	78	4E	N	110	6E	n
15	F	[SHIFT IN]	47	2F	/	79	4F	O	111	6F	o
16	10	[DATA LINK ESCAPE]	48	30	0	80	50	P	112	70	p
17	11	[DEVICE CONTROL 1]	49	31	1	81	51	Q	113	71	q
18	12	[DEVICE CONTROL 2]	50	32	2	82	52	R	114	72	r
19	13	[DEVICE CONTROL 3]	51	33	3	83	53	S	115	73	s
20	14	[DEVICE CONTROL 4]	52	34	4	84	54	T	116	74	t
21	15	[NEGATIVE ACKNOWLEDGE]	53	35	5	85	55	U	117	75	u
22	16	[SYNCHRONOUS IDLE]	54	36	6	86	56	V	118	76	v
23	17	[ENG OF TRANS. BLOCK]	55	37	7	87	57	W	119	77	w
24	18	[CANCEL]	56	38	8	88	58	X	120	78	x
25	19	[END OF MEDIUM]	57	39	9	89	59	Y	121	79	y
26	1A	[SUBSTITUTE]	58	3A	:	90	5A	Z	122	7A	z
27	1B	[ESCAPE]	59	3B	;	91	5B	[123	7B	{
28	1C	[FILE SEPARATOR]	60	3C	<	92	5C	\	124	7C	
29	1D	[GROUP SEPARATOR]	61	3D	=	93	5D]	125	7D	}
30	1E	[RECORD SEPARATOR]	62	3E	>	94	5E	^	126	7E	~
31	1F	[UNIT SEPARATOR]	63	3F	?	95	5F	_	127	7F	[DEL]

Επεκτάσεις ASCII κώδικα

Σημαντικό μειονέκτημα του κώδικα ASCII είναι ότι δεν είναι δυνατή η αναπαράσταση χαρακτήρων άλλου αλφαβήτου εκτός από το λατινικό. Γι' αυτό το λόγο χρησιμοποιήθηκε και το όγδοο bit για την κωδικοποίηση χαρακτήρων και δημιουργήθηκαν **επεκτάσεις του κώδικα**.

Έτσι έγινε δυνατή η αναπαράσταση 128 επιπλέον χαρακτήρων ($2^8=256$), που χρησιμοποιήθηκαν από κάθε χώρα για την αναπαράσταση γραμμάτων του τοπικού αλφαβήτου, καθώς και άλλων χαρακτήρων (π.χ. μαθηματικά σύμβολα).

Κώδικας χαρακτήρων Unicode



Ο κώδικας **Unicode** είναι ένα παγκόσμιο σύνολο χαρακτήρων (universal character set), δηλ. ένα διεθνές πρότυπο που παρέχει τη δυνατότητα κωδικοποίησης όλων των χαρακτήρων των σημαντικότερων γλωσσών του κόσμου.

Προτείνει έναν μοναδικό αριθμό (code point) για κάθε χαρακτήρα, ανεξάρτητα από το λειτουργικό σύστημα, το λογισμικό και τη φυσική γλώσσα. Η τιμή του code point παριστάνεται με το πρόθεμα U+ ακολουθούμενο από τη δεκαεξαδική μορφή της θέσης του.

Κωδικοποιήσεις Unicode

UTF-8 (8-bit Unicode Transformation Format)

- 1 byte για τους 128 χαρακτήρες ASCII
- 2 bytes για τους χαρακτήρες των υπόλοιπων γλωσσών
- 3 bytes για τις υπόλοιπες «περίεργες» γλώσσες
- 4 bytes για τους συμπληρωματικούς χαρακτήρες

UTF-16

- 2 bytes για κάθε χαρακτήρα των φυσικών γλωσσών
- 4 bytes για τους συμπληρωματικούς χαρακτήρες.

UTF-32

- 4 bytes για κάθε διαφορετικό χαρακτήρα.

character	encoding	bits
A	UTF-8	01000001
A	UTF-16	00000000 01000001
A	UTF-32	00000000 00000000 00000000 01000001
あ	UTF-8	11100011 10000001 10000010
あ	UTF-16	00110000 01000010
あ	UTF-32	00000000 00000000 00110000 01000010

3.2 Μέθοδοι συμπίεσης δεδομένων



ΕΙΣΗΓΗΤΗΣ: ΣΠΑΧΟΣ ΚΥΡΙΑΚΟΣ – ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ

Συμπίεση δεδομένων

Συμπίεση δεδομένων είναι η διαδικασία μείωσης του όγκου αρχείων διαφόρων ειδών από το αρχικό τους μέγεθος.

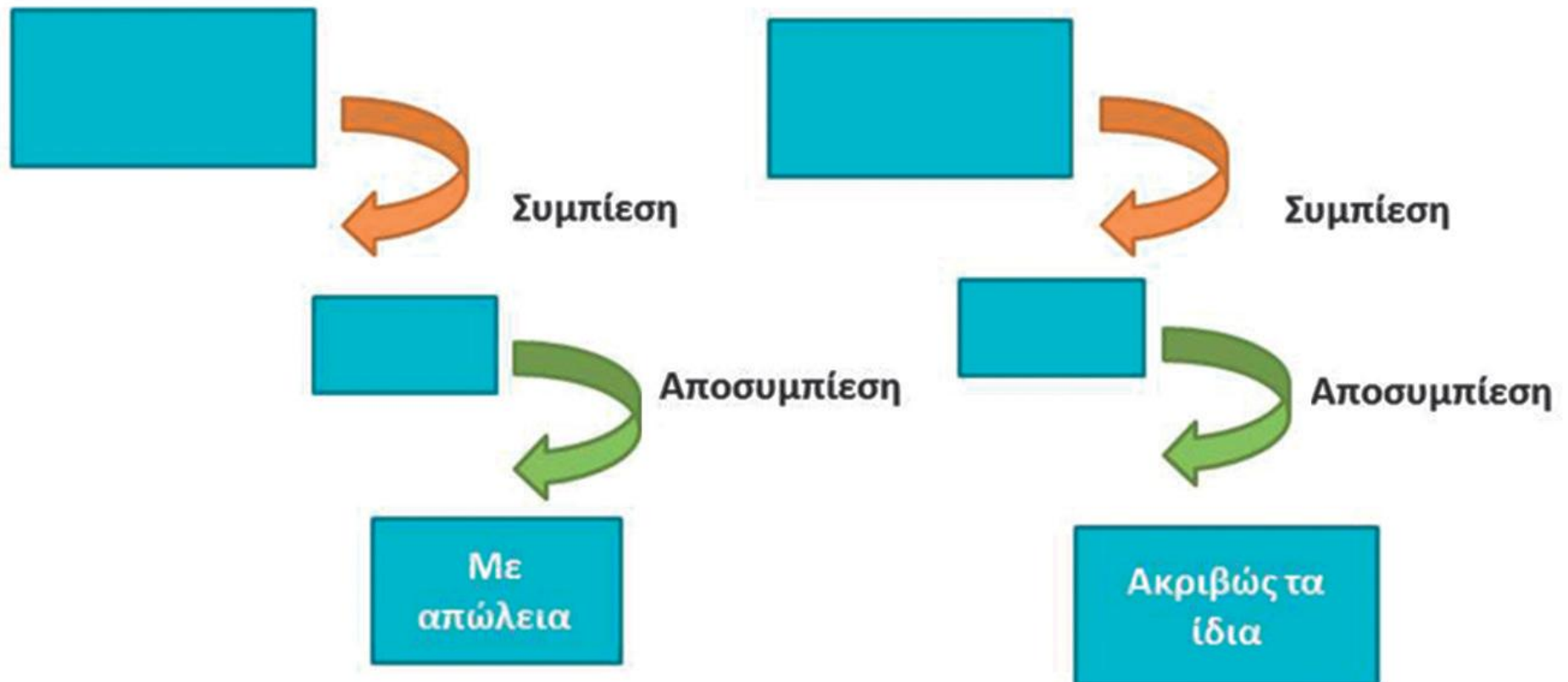
Στόχος της συμπίεσης είναι η ταχύτερη μεταφορά μέσω δικτύων και η εξοικονόμηση αποθηκευτικού χώρου.

Αποσυμπίεση δεδομένων είναι η επαναφορά των δεδομένων στην αρχική κατάσταση.



Μέθοδοι συμπίεσης

Απωλεστική συμπίεση - Μη απωλεστική συμπίεση



Απωλεστική συμπίεση:

Τα δεδομένα μετά τη συμπίεση και την αποσυμπίεση παρουσιάζουν απώλεια

αρχεία ήχου, εικόνες, βίντεο

Μη απωλεστική συμπίεση:

Τα αρχικά δεδομένα και τα δεδομένα μετά τη συμπίεση και την αποσυμπίεση είναι ακριβώς τα ίδια

βάσεις δεδομένων, πηγαίος κώδικας, emails, έγγραφα, λογιστικά φύλλα

Μέθοδοι συμπίεσης

Μέθοδοι Συμπίεσης

Απωλεστική συμπίεση

JPEG

MP3

MPEG

Μη απωλεστική συμπίεση

Run Length

Lempel-Ziv

Huffman

Αλγόριθμος συμπίεσης Ziv-Lempel

Έστω το παρακάτω μήνυμα:

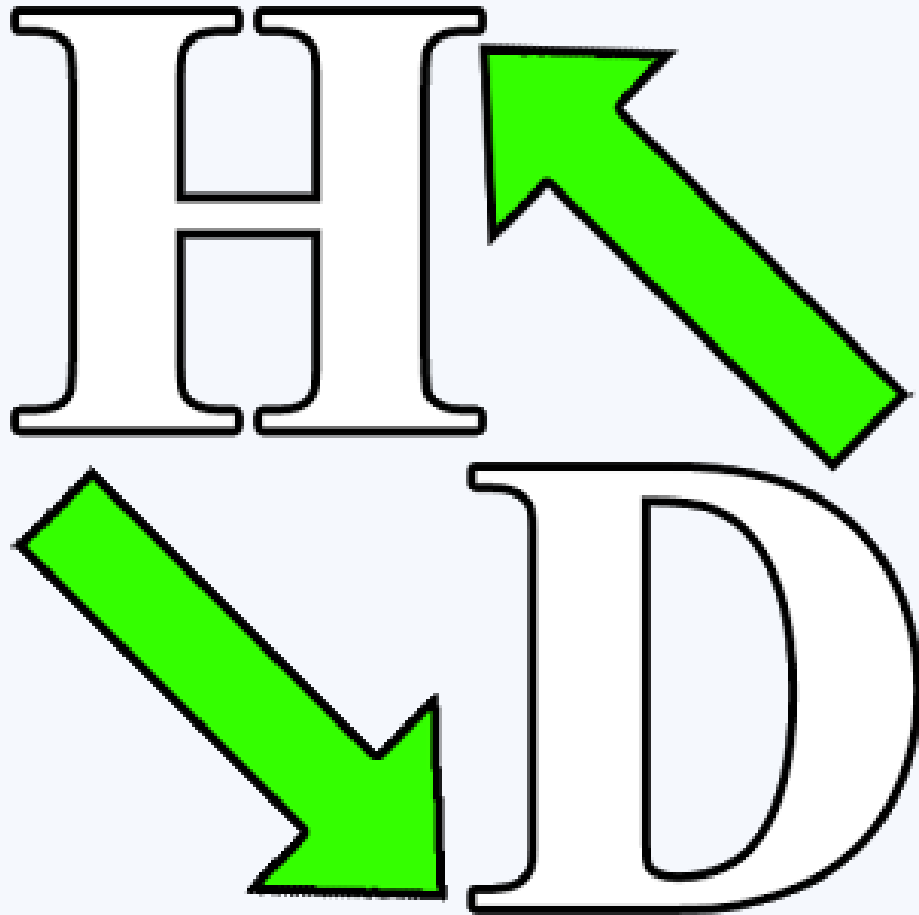
VJGDNQMYLH-KW-VJGDNQMYLH-ADXSGFO-VJGDNQMYLH- ADXSGF-
VJGDNQMYLH-EWADXSGF

Καθώς μεταδίδουμε αυτό το μήνυμα - για να γλυτώσουμε κόπο - μπορούμε να πούμε πόσο πριν και πόσους χαρακτήρες περιέχει το επαναλαμβανόμενο τμήμα δηλ., πρέπει να πούμε κάτι σαν “πήγαινε πίσω 27 χαρακτήρες και ξεκινώντας από αυτό το σημείο αντίγραψε 8 χαρακτήρες”

Με αυτή την «τεχνική» ένα ολόκληρο κομμάτι κειμένου μπορεί να αντικατασταθεί από δύο αριθμούς. Ο πρώτος δείχνει πόσα σύμβολα πρέπει να πάμε πίσω και ο δεύτερος το μήκος του κειμένου που θα αντιγράψουμε.

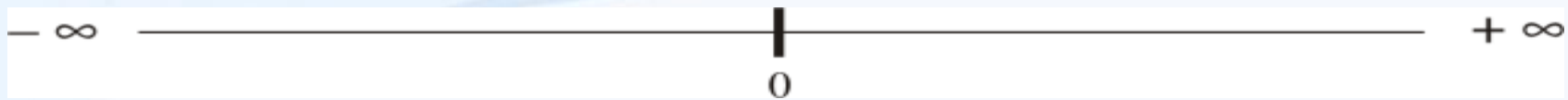
Ο αλγόριθμος αυτός χρησιμοποιείται κυρίως σε αρχεία κειμένου και μειώνει το μέγεθος στο $\frac{1}{4}$ του αρχικού.

3.3 Αναπαράσταση ακεραίων αριθμών



ΕΙΣΗΓΗΤΗΣ: ΣΤΑΧΟΣ ΚΥΡΙΑΚΟΣ - ΠΕ19 ΠΛΗΡΟΦΟΡΙΚΗΣ

Αναπαράσταση ακεραίων



Δεν υπάρχει υπολογιστής που να μπορεί να αποθηκεύσει όλους τους ακέραιους σε παραπάνω διάστημα τιμών => θα χρειαζόταν άπειρο πλήθος μπιτ, δηλ. άπειρη αποθηκευτική ικανότητα.

Για την αποδοτικότερη χρήση της μνήμης των υπολογιστών έχουν αναπτυχθεί δύο μεγάλες κατηγορίες αναπαράστασης ακεραίων: προσημασμένοι και μη προσημασμένοι ακέραιοι.



Μη προσημασμένοι ακέραιοι

- Ένας **μη προσημασμένος ακέραιος** είναι ένας ακέραιος χωρίς πρόσημο που μπορεί να πάρει τιμές από το 0 μέχρι το θετικό άπειρο
- Επειδή δεν υπάρχει υπολογιστής που να μπορεί να αναπαραστήσει όλους τους ακέραιους σε αυτό το διάστημα τιμών, ορίζεται μια σταθερά που ονομάζεται **μέγιστος μη προσημασμένος ακέραιος** και έτσι ένας μη προσημασμένος ακέραιος μπορεί να πάρει τιμές από το 0 μέχρι αυτή τη σταθερά.
- Ο μέγιστος μη προσημασμένος ακέραιος εξαρτάται από τον **αριθμό N των bit που χρησιμοποιεί ο υπολογιστής για την αναπαράσταση** ενός ακέραιου.
- Με τη χρήση n bit για αναπαράσταση ο υπολογιστής μπορεί να αναπαραστήσει 2^n διαφορετικούς ακεραίους ή αλλιώς τους μη προσημασμένους ακεραίους από **$0 \dots (2^n - 1)$** .

Μη προσημασμένοι ακέραιοι

Δυαδικά ψηφία (bits)	Μεγαλύτερος μη προσημασμένος ακέραιος
8	$2^8-1 = 255$
16	$2^{16}-1 = 65.535$
32	$2^{32}-1 = 4.294.967.295$
64	$2^{64}-1 = 18.446.744.073.709.551.615$
128	$2^{128}-1 = 340.282.366.920.938.463.463.374.607.431.768.211.455$

Η αποθήκευση μη προσημασμένων ακέραιων, σε έναν υπολογιστή που χρησιμοποιεί n bits, είναι μια απλή διαδικασία:

1. Ο αριθμός μετατρέπεται στο δυαδικό σύστημα.
2. Αν το πλήθος των bit, που απαιτούνται για τον αριθμό, είναι μικρότερο από n τότε προστίθενται μηδενικά στα αριστερά του δυαδικού αριθμού ώστε να υπάρχουν συνολικά n bits.

Μη προσημασμένοι ακέραιοι

Αποθήκευση μη προσημασμένων ακεραίων σε δύο διαφορετικούς υπολογιστές με 8 και 16 bits

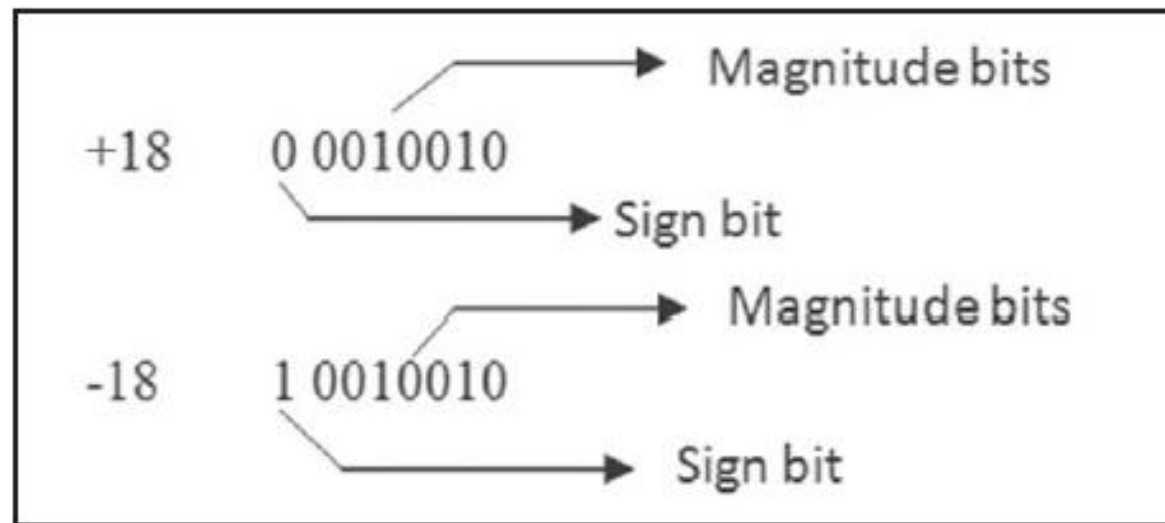
Δεκαδικός	Δέσμευση 8 μπιτ	Δέσμευση 16 μπιτ
7	00000111	0000000000000111
234	11101010	0000000011101010
258	Υπερχείλιση	0000000100000010
24.760	Υπερχείλιση	0110000010111000
1.245.678	Υπερχείλιση	Υπερχείλιση

Αν ο ακέραιος προς αποθήκευση είναι μεγαλύτερος από το μέγιστο μη προσημασμένο τότε έχουμε μια κατάσταση που ονομάζεται **υπερχείλιση**

Προσημασμένοι ακέραιοι σε μορφή πρόσημου - μέτρου

Το πρόσημο ενός αριθμού αναπαρίσταται ξεχωριστά από το μέτρο του.

- Το ψηφίο μέγιστης σημαντικότητας (MSB), είναι το ψηφίο πρόσημο (*sign bit*), με τη σύμβαση ότι το 0 είναι για τους θετικούς ακεραίους και το 1 για τους αρνητικούς ακεραίους.
- Τα υπόλοιπα $n-1$ bits αντιπροσωπεύουν το μέτρο (απόλυτη τιμή) (*magnitude bits*) του ακεραίου..



Προσημασμένοι ακέραιοι σε μορφή πρόσημου - μέτρου

- Στην αναπαράσταση πρόσημου και μεγέθους, το τελευταίο αριστερά μπιτ καθορίζει το πρόσημο του αριθμού.
 - Αν είναι 0, ο αριθμός είναι θετικός
 - Αν είναι 1, ο αριθμός είναι αρνητικός
- Υπάρχουν δύο μηδενικά: ένα θετικό και ένα αρνητικό. Η μορφή τους σε μια δέσμευση 8 μπιτ είναι η εξής:
 - +0 -> 00000000
 - -0 -> 10000000

Πλήθος μπιτ	Διάστημα τιμών
8	-127 ... -0 +0 ... +127
16	-32.767 ... -0 +0 ... +32.767
32	-2.147.483.647 ... -0 +0 .. + 2.147.483.647

Αποθήκευση Προσημασμένων σε μορφή πρόσημου – μέτρου

- Ο αριθμός μετατρέπεται στο δυαδικό σύστημα, το πρόσημο αγνοείται.
- Αν το πλήθος των μπιτ είναι μικρότερο από $n-1$, προστίθενται μηδενικά στα αριστερά του αριθμού ώστε να υπάρχει ένα σύνολο από $n-1$ μπιτ.
- Αν ο αριθμός είναι θετικός, προστίθεται στα αριστερά ένα μηδενικό (ώστε να έχουμε σύνολο n μπιτ).
- Αν ο αριθμός είναι αρνητικός, προστίθεται στα αριστερά η μονάδα (ώστε και πάλι το σύνολο να είναι n μπιτ).

Προσημασμένοι ακέραιοι σε μορφή πρόσημου - μέτρου

Αποθηκεύστε τον αριθμό -258 σε μια θέση μνήμης 16 bit με την αναπαράσταση πρόσημου και μεγέθους.

Λύση

- Πρώτα μετατρέπουμε την απόλυτη τιμή του αριθμού στο δυαδικό σύστημα και προκύπτει ο αριθμός 100000010
- Προσθέτουμε έξι 0 ώστε να έχουμε σύνολο $n-1$ (15) μπιτ και έτσι προκύπτει ο αριθμός 000000100000010
- Επειδή ο αριθμός είναι αρνητικός, προσθέτουμε ένα 1 αριστερά, το οποίο φαίνεται με έντονη γραφή.
- Το αποτέλεσμα είναι **1000000100000010**

Προσημασμένοι ακέραιοι σε μορφή πρόσημου - μέτρου

Δεκαδικός	Δέσμευση 8 μπιτ	Δέσμευση 16 μπιτ
+7	00000111	00000000000000111
-124	11111100	1000000001111100
+258	Υπερχείλιση	0000000100000010
-24760	Υπερχείλιση	1110000010111000

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 1

Το ψηφίο μέγιστης σημαντικότητας (MSB), είναι το ψηφίο προσήμου, με τη σύμβαση ότι το 0 είναι για τους θετικούς ακεραίους και το 1 για τους αρνητικούς ακεραίους.

Τα υπόλοιπα $n-1$ bits αντιπροσωπεύουν το μέτρο του ακεραίου, ως εξής:

- αν ο αριθμός είναι θετικός, το μέτρο του δίδεται από τα υπόλοιπα $n-1$ bits.
- αν ο αριθμός είναι αρνητικός, το μέτρο του δίνεται από το συμπλήρωμα ως προς 1 (αντίστροφο) των υπολοίπων $n-1$ bits. Το συμπλήρωμα ως προς 1 ενός δυαδικού αριθμού βρίσκεται εύκολα αν αντικατασταθούν όλα τα 1 του αριθμού με 0 και όλα τα 0 με 1.

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 1

- Στην αναπαράσταση συμπληρώματος ως προς ένα υπάρχουν δύο μηδενικά: ένα θετικό και ένα αρνητικό.

Σε μια αναπαράσταση 8 bit αυτό έχει ως εξής:

+0 -> 00000000

-0 -> 11111111

- Διάστημα τιμών αναπαράστασης $-(2^{n-1} - 1) \dots +(2^{n-1} - 1)$

Πλήθος μπιτ	Διάστημα τιμών
8	-127 ... -0 +0 ... +127
16	-32.767 ... -0 +0 ... +32.767
32	-2.147.483.647 ... -0 +0 .. + 2.147.483.647

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 1

Διαδικασία αποθήκευσης:

- Η απόλυτη τιμή του ακέραιου αριθμού μετατρέπεται στο δυαδικό σύστημα.
- Προστίθενται μηδενικά στα αριστερά του αριθμού ώστε να υπάρχει ένα σύνολο από n bit.
- Αν ο αριθμός είναι θετικός, δε χρειάζεται άλλη ενέργεια
- Αν ο αριθμός είναι αρνητικός, κάθε bit αντικαθίσταται από το συμπλήρωμά του (τα 0 γίνονται 1 και τα 1 γίνονται 0)

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 1

Αποθηκεύστε τον αριθμό -258 σε μια θέση μνήμης 16 bit με την αναπαράσταση συμπληρώματος ως προς 1

Λύση

- Πρώτα μετατρέπουμε την απόλυτη τιμή του αριθμού στο δυαδικό σύστημα και προκύπτει ο αριθμός 100000010
- Προσθέτουμε επτά 0 ώστε να έχουμε σύνολο n (16) bit και έτσι προκύπτει ο αριθμός 0000000100000010
- Επειδή ο αριθμός είναι αρνητικός, αντικαθιστούμε κάθε 0 με 1 και κάθε 1 με μηδέν
- Το αποτέλεσμα είναι 1111111011111101

Μετατροπή δυαδικού προσημασμένου από μορφή συμπληρώματος ως προς 1 σε δεκαδικό.

Τα βήματα για την ερμηνεία μιας δυαδικής αναπαράστασης συμπληρώματος ως προς ένα στο δεκαδικό σύστημα είναι τα ακόλουθα:

- Αν το τελευταίο αριστερά μπιτ είναι 0 (θετικός αριθμός),
 - ❖ Μετατρέπουμε ολόκληρο τον αριθμό από το δυαδικό στο δεκαδικό σύστημα.
 - ❖ Τοποθετούμε θετικό πρόσημο (+) μπροστά από τον αριθμό.
- Αν το τελευταίο αριστερά μπιτ είναι 1 (αρνητικός αριθμός),
 - ❖ Αντικαθιστούμε τον αριθμό με το συμπλήρωμά του (αλλάζουμε όλα τα 0 σε 1, και το αντίστροφο).
 - ❖ Μετατρέπουμε ολόκληρο τον αριθμό από το δυαδικό στο δεκαδικό σύστημα.
 - ❖ Τοποθετούμε μπροστά από τον αριθμό αρνητικό πρόσημο (-).

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 2

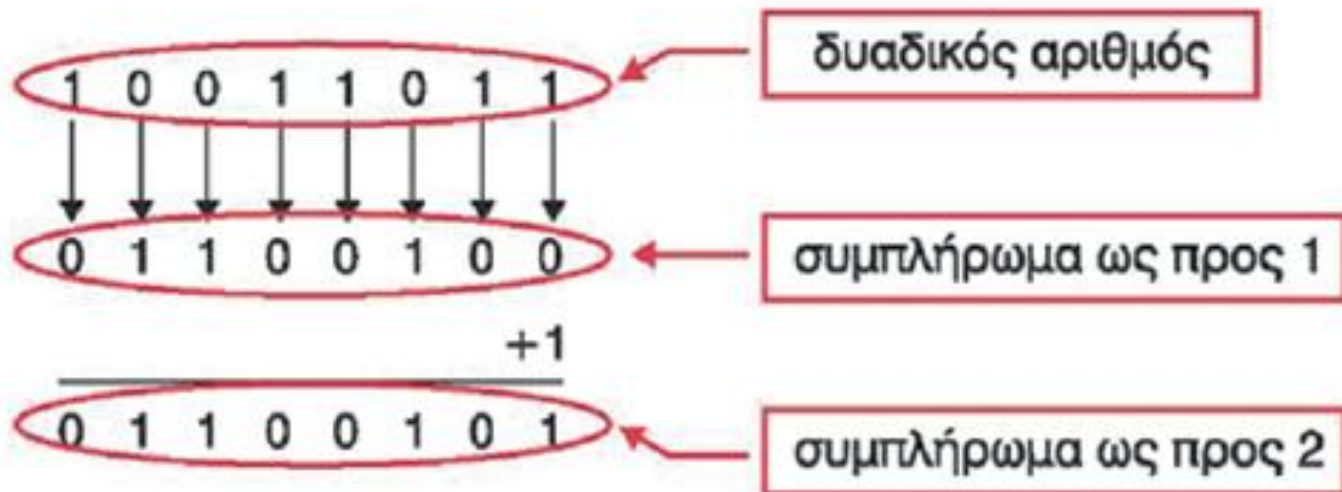
- Η αναπαράσταση συμπληρώματος ως προς ένα έχει δύο μηδέν (+0 και -0), γεγονός που μπορεί να προκαλέσει σύγχυση σε υπολογισμούς
- Επίσης, αν προσθέσουμε έναν αριθμό με το συμπλήρωμά του ως προς 1 (π.χ. +4 και -4) σε αυτή την αναπαράσταση, παίρνουμε ως αποτέλεσμα αρνητικό μηδέν (-0) αντί για θετικό (+0)
- Η αναπαράσταση συμπληρώματος ως προς δύο λύνει όλα αυτά τα προβλήματα

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 2

Η παράσταση συμπληρώματος ως προς δύο είναι ο πιο συνηθισμένος τρόπος αναπαράστασης ακεραίων κατά τη διάρκεια επεξεργασίας από έναν υπολογιστή.

- Όπως και στις προηγούμενες παραστάσεις, αν το MSB του αριθμού είναι 0, ο αριθμός είναι θετικός και το μέτρο του δίδεται από τα υπόλοιπα $n-1$ bits.
- Εάν το MSB του αριθμού είναι 1, τότε ο αριθμός είναι αρνητικός. Για να βρούμε το μέτρο του αριθμού, πρέπει να υπολογίσουμε το συμπλήρωμα ως προς 2 και των n ψηφίων του (δηλαδή λαμβάνουμε υπόψη και το πρόσημο). Το συμπλήρωμα ως προς 2 ενός δυαδικού αριθμού βρίσκεται, εάν αντικαταστήσουμε το 0 με 1 και το 1 με 0 (συμπλήρωμα ως προς 1) και στη συνέχεια προσθέσουμε 1.

Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 2



Προσημασμένοι ακέραιοι σε μορφή συμπληρώματος ως προς 2

- Το συμπλήρωμα ως προς δύο αποτελεί σήμερα τον πιο συνηθισμένο, τον πιο σημαντικό, και τον πιο ευρέως χρησιμοποιούμενο τρόπο αναπαράστασης ακεραίων
- Διάστημα τιμών: $-(2^n-1) \dots + (2^{n-1}-1)$
- Υπάρχει μόνο ένα μηδενικό

Πλήθος μπιτ	Διάστημα τιμών
8	-128 ... 0 ... +127
16	-32.768 ... 0 ... +32.767
32	-2.147.483.648 ... 0 .. + 2.147.483.647

Αποθήκευση προσημασμένου σε μορφή συμπληρώματος ως προς 2

- ❖ Η απόλυτη τιμή του ακεραίου μετατρέπεται στο δυαδικό σύστημα.
- ❖ Αν το πλήθος των bit είναι μικρότερο από n , προστίθενται μηδενικά στα αριστερά του αριθμού ώστε να υπάρχει ένα σύνολο από n μπιτ.
- ❖ Αν το πρόσημο είναι θετικό, δε χρειάζεται καμία άλλη ενέργεια.
- ❖ Αν το πρόσημο είναι αρνητικό, μένουν ως έχουν όλα τα δεξιότερα 0 και το πρώτο 1. Τα υπόλοιπα bit αντικαθίστανται από το συμπλήρωμά τους.

Αποθήκευση προσημασμένου σε μορφή συμπληρώματος ως προς 2

Αποθηκεύστε τον αριθμό -40 σε μια θέση μνήμης 16 bit με την αναπαράσταση συμπληρώματος ως προς δύο

Λύση

- ❖ Πρώτα μετατρέπουμε τον αριθμό 40 στο δυαδικό σύστημα (101000).
- ❖ Προσθέτουμε δέκα 0 ώστε να έχουμε σύνολο n (16) bit (0000000000101000).
- ❖ Ο αριθμός είναι αρνητικός, οπότε αφήνουμε τα δεξιότερα 0 μέχρι το πρώτο 1 (και το 1) ως έχουν, και αντικαθιστούμε τα υπόλοιπα bit με το συμπλήρωμά τους.
- ❖ Το αποτέλεσμα είναι 1111111111011000.

Μετατροπή δυαδικού προσημασμένου από μορφή συμπληρώματος ως προς 2 σε δεκαδικό.

Τα βήματα για την ερμηνεία μιας δυαδικής αναπαράστασης συμπληρώματος ως προς δύο στο δεκαδικό σύστημα είναι τα ακόλουθα:

- ❖ Αν το τελευταίο αριστερά μπιτ είναι 0 (θετικός αριθμός) μετατρέπουμε ολόκληρο τον αριθμό από το δυαδικό στο δεκαδικό σύστημα.
- ❖ Τοποθετούμε θετικό πρόσημο (+) μπροστά από τον αριθμό
- ❖ Αν το τελευταίο αριστερά μπιτ είναι 1 (αρνητικός αριθμός)
- ❖ Αφήνουμε τα δεξιότερα μηδενικά bit μέχρι το πρώτο 1 (μαζί με αυτό) ως έχουν. Αντικαθιστούμε τα υπόλοιπα μπιτ με το συμπλήρωμά τους.
- ❖ Μετατρέπουμε ολόκληρο τον αριθμό από το δυαδικό στο δεκαδικό σύστημα.
- ❖ Τοποθετούμε μπροστά από τον αριθμό αρνητικό πρόσημο (-).

Υπερχείλιση

Με όποιον τρόπο κι αν παριστάνονται οι αριθμοί σε ένα υπολογιστή, υπάρχει πάντα ένα **ανώτερο** και ένα **κατώτερο** όριο στο μεγεθός τους. Τα όρια εξαρτώνται από:

- τον τρόπο παράστασης που χρησιμοποιείται
- τον αριθμό των δυαδικών ψηφίων (*bits*) που διατίθεται για τους αριθμούς αυτούς

Ο όρος **υπερχείλιση (overflow)** χρησιμοποιείται για να περιγράψει την περίπτωση που το αποτέλεσμα μιας πράξης βρίσκεται εκτός αυτών των ορίων (εκτός του εύρους τιμών).

Τα περισσότερα υπολογιστικά συστήματα χρησιμοποιούν διαφορετικό πλήθος ψηφίων για την παράσταση κάθε τύπου αριθμών, έτσι ώστε να μειώσουν την πιθανότητα σφαλμάτων από υπερχείλιση

Αναπαράσταση αριθμών στη C

Type	Bits	Range
int	16	-32768 to 32767
unsigned int	16	0 to 65535
signed int	16	-32768 to 32767
short int	16	-32768 to 32767
unsigned short int	16	0 to 65535
signed short int	16	-32768 to 32767
long int	32	-2147483648 to 2147483647
unsigned long int	32	0 to 4294967295
signed long int	32	-2147483648 to 2147483647
float	32	3.4E-38 to 3.4E+38
double	64	1.7E-308 to 1.7E+308
long double	80	3.4E-4932 to 3.4E+4932
char	8	-128 to 127
unsigned char	8	0 to 255
signed char	8	-128 to 127

Δραστηριότητες



1. Αναζητήστε πληροφορίες για την οικογένεια προτύπων ISO 8859 <https://goo.gl/LLqPyT> και τις κωδικοσελίδες των Windows <https://goo.gl/mlMzxs>. Περιηγηθείτε στον Πίνακα ASCII (<http://goo.gl/DDvnhgy>).
2. Αναζήτησε και χρησιμοποίησε online μετατροπείς (π.χ. Ascii Text to Binary Converter).
3. Αναζητήστε τον εκτεταμένο κώδικα ASCII για υποστήριξη ελληνικών.
4. Αναζητήστε πληροφορίες για το ψηφίο ισοτιμίας (parity digit) και πως συμβάλει στην σωστή μετάδοση της πληροφορίας
5. Δείτε το παρακάτω παιχνίδι σχετικά με τη λειτουργία του ψηφίου ισοτιμίας, πατώντας [εδώ](#)
6. Επίσης στο [video](#) μπορείτε να δείτε τι είναι το ψηφίο ισοτιμίας
7. Πώς δηλώνουμε στην ετικέτα HTML <meta> την κωδικοποίηση μιας ιστοσελίδας; Αναζήτησε πληροφορίες και δείτε το σύνδεσμο (<http://goo.gl/6AvRpR>, <http://goo.gl/taq07r>).