

MÁSODIK GYAKORLAT

SZÁMRENDSZEREK, KÓDOLÁS

A feladat elvégzése során a következőket fogjuk gyakorolni:

- Átváltás tízes, tizenhatos és kettes számrendszerek között.
- Előjeles egész számok és tört értékek fixpontos ábrázolása.
- Alfánumerikus jelek kódolása.
- Műveletek logikai értékekkel, Boole algebra.

A feladat megoldása hozzávetőlegesen 80 percet vesz igénybe.

Nem-negatív egészek átírása más számrendszerre

A 10-es számrendszerben megadott (pl. 2013) értéket írjuk át 2-es számrendszerbeli alakra! Az eredményt adjuk meg 16 pozíción (azaz 2 bájtton)!

Megoldás: sorozatosan elvégezzük a megadott értékből indulva az alapszámmal (most a 2-vel) való osztást és két oszlopba rendezve feljegyezzük a lefelé kerekített eredményt és a maradékot. Ezt mindaddig tesszük, míg az eredmény 0-vá nem válik. A maradékok fordított sorrendű jelsorozataként kapjuk a megadott érték új alapszámú jelsorozatát.

2013	maradék
1006	1
503	0
251	1
125	1
62	1
31	0
15	1
7	1
3	1
1	1
0	1

Tehát a 2-es számrendszerben a **vezető nullákkal** kiegészített bináris alak:

0 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1

Valóban

$$2^{10}+2^9+2^8+2^7+2^6+2^4+2^3+2^2+2^0 = 1024+512+256+128+64+16+8+4+1 = 2013$$

ÖNÁLLÓ FELADAT

ÍRJUK ÁT AZ ADOTT ÉRTÉKET 8-AS SZÁMRENDSZERBE IS, ÉS VÉGEZZÜK EL AZ ELLENŐRZÉST!

A 16-os számrendszerben a 0..15 értékek jelölésére a 0, 1, 2, ... 9, A, B, C, D, E, F írásjeleket használjuk. A fenti maradékos osztási algoritmus analógiájára írjuk fel az adott értéket a 16-os (hexadecimális) számrendszer segítségével 4 pozíción!

2013	maradék
125	13
7	13
0	7
0	0

Tehát a keresett hexadecimális forma: 07DD (a D jel a 13 decimális értéket kódolja).

A gyak1.xls második füzetlapján látjuk a decimális – bináris – hexadecimális átváltó táblázatot. Ennek segítségével a 07DD-vel hexadecimálisan adott érték gyorsan átírható binárisra:

0 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1

Ez pedig éppen az a bitsorozat, amelyet a 10-es számrendszerű alakból bináriszá váló átírásnál az előzőekben kaptunk.

ÖNÁLLÓ FELADAT

A FENTI MINTÁRA VÉGEZZÜK EL SZÁMOLÓGÉP HASZNÁLATA NÉLKÜL A SZÜLETÉSI DÁTUMUNKBÓL KÉPZETT NÉGYJEGYŰ (PÉLDÁUL: 1992. MÁRCIUS 12. ESETÉN 9212) DECIMÁLIS SZÁMOT KETTES ÉS TÍZENHATOS SZÁMRENDSZERBE VALÓ ÁTÍRÁSÁT IS!

Előjeles egészek átírása kettes számrendszerbe

Az előjeles számok tárolásánál a legnagyobb helyértékű bitet feláldozzuk az előjel kódolására a következő szisztémával.

Ha az előjelbit 0, akkor a kódolt szám pozitív és a kódolása az előbb megismert módon történik.

Negatív értékek kódolására pedig az ún. 2-es komplementst használjuk a következő módon:

1. Írjuk fel a negatív szám abszolút-értékének bináris alakját!
2. Fordítsunk meg ellenkezőjére minden bitet (negálás)!
3. Adjunk hozzá 1-t!

A kapott kód a negatív értékünk 2-es komplement kódja. Érdemes megjegyezni, hogy egy 1-es előjelbittel rendelkező negatív érték abszolút-értékét pontosan az 1-3 lépések megismétlésével kapjuk meg.

Állítsuk elő a -2013 érték 2-es komplement kódját!

1. lépés (2013 kódja):

0 0 0 0 0 1 1 1 1 1 0 1 1 1 0 1

2. lépés (negálás):

1 1 1 1 1 0 0 0 0 0 1 0 0 0 1 0

3. lépés (1 hozzáadása):

1 1 1 1 1 0 0 0 0 0 1 0 0 0 1 1

ÖNNÁLLÓ FELADATOK:

ELLENŐRIZZÜK, HOGY A (-2013) KÓDJÁBÓL KIINDULVA VÉGREHAJTJUK AZ 1-3. LÉPÉSEKET, AKKOR A NEGATÍV SZÁM ABSZOLÚT-ÉRTÉKÉNEK (AZAZ A 2013-NAK) A KÓDJÁHOZ JUTUNK!

A GYAK1.XLS SEGÍTSÉGÉVEL ELLENŐRIZZÜK AZ EDDIG KÉZZEL VÉGZETT SZÁMÍTÁSAINKAT!

Fixpontos számábrázolás

Fixpontos számábrázolásnál a tizedes (kettes) elválasztójel helye rögzített. Ilyen számábrázolást alkalmaznak például a pénztárgépekben, ahol csak két tizedes jegy pontosan jelenít meg a fizetendő összeget például: 18325,40 Ft. Természetesen adott mennyiségű szám-

jegy esetében az elválasztójel helye egyszerre van hatással az ábrázolható számok nagyságára és az ábrázolás pontosságára.

Feladat: ábrázoljuk a 2013,2013 decimális számot 32 bites, fixpontos, kettes számrendszerbeli alakban. Az egész és törtreszt egyaránt ábrázoljuk 16 biten a következő ábrán jelzett módon.



Megoldás: Az egészrész átalakításánál pontosan úgy kell eljárunk, ahogy azt az első feladatban már megismertük. A törtresz átalakítása is hasonló, azzal a különbséggel, hogy nem osztani, hanem szorozni kell, és az egészek közé „felcsúszó” számjegyek adják az átalakított értéket.

egész	0,2013
0	,4026
0	,8052
1	,6104
1	,2208
0	,4416
0	,8832
1	,7664
1	,5328
1	,0656
0	,1312
0	,2624
0	,5248
1	,0496
0	,0992
0	,1984
0	,3968

(csak a törtreszeket szorozzuk)

Az eljárást egészen addig folytatjuk, amíg eredményül nullát nem kapunk, illetve el nem érjük a kívánt pontosságot. A pirossal jelzett egészrészek fentről lefelé haladva adják az átalakított törtresz kettes számrendszerbeli alakját: 0011001110001000

$$2^{-3}+2^{-4}+2^{-7}+2^{-8}+2^{-9}+2^{-13}\approx 0,2012939$$

Végül egymás mellé írjuk az egész és törtrészeket:

$$2013,2013_{(10)}=00000111110111010011001110001000_{(2)}$$

ÖNNÁLLÓ FELADAT:

A FENTI MINTÁRA VÉGEZZÜK EL A SZÜLETÉSI DÁTUMUNKBÓL KÉPZETT NYOLCJEGYŰ (PÉLDÁUL: 1992. MÁRCIUS 12. ESETÉN 1992,0312) 2-ES SZÁMRENDSZERBE VALÓ ÁTÍRÁSÁT, 16 BIT EGÉSZRÉSZT ÉS 16 BIT TÖRTRESZT TARTALMAZÓ FIXPONTOS ÁBRÁZOLÁSSAL. A GYAKI.XLS SEGÍTSÉGÉVEL ELLENŐRIZZÜK AZ ELVÉGZETT SZÁMÍTÁSOKAT!

Alfanumerikus jelek (szöveg) kódolása

A számítógép nemcsak a számokat, hanem más objektumokat, például a betűket, írásjeleket és egyéb szimbólumokat is binárisan tárol. Ehhez természetesen szükség van valamilyen kódrendszerre, összerendelő táblázatra, amely az előbb említett szimbólumokat és a hozzájuk tartozó számértéket tartalmazza.

Az évek során több ilyen kódolási szabvány (kódlap) született. Ezek közül sokáig egyeduralkodónak számított az egy bájt széles ASCII (American Standard Code for Information Interchange) kódolás, amely összesen 128 darab (00-7F) szimbólumot: az angol ábécé betűit, számokat, írásjeleket és egyéb vezérlőkaraktereket tartalmaz.

Az ASCII hiányossága, hogy nem alkalmas más nemzetek speciális írásjeleinek megjelenítésére (például hiányoznak az ékezetes magyar betűk). A problémát kezdetben úgy orvosolták, hogy az ASCII kódtábla üresen maradt (80-FF) részét kitöltve, a különböző nemzetek igényeihez igazodó kódlapokat hoztak létre (ISO8-859-1, ISO8859-2, CP852, windows1250 stb.).

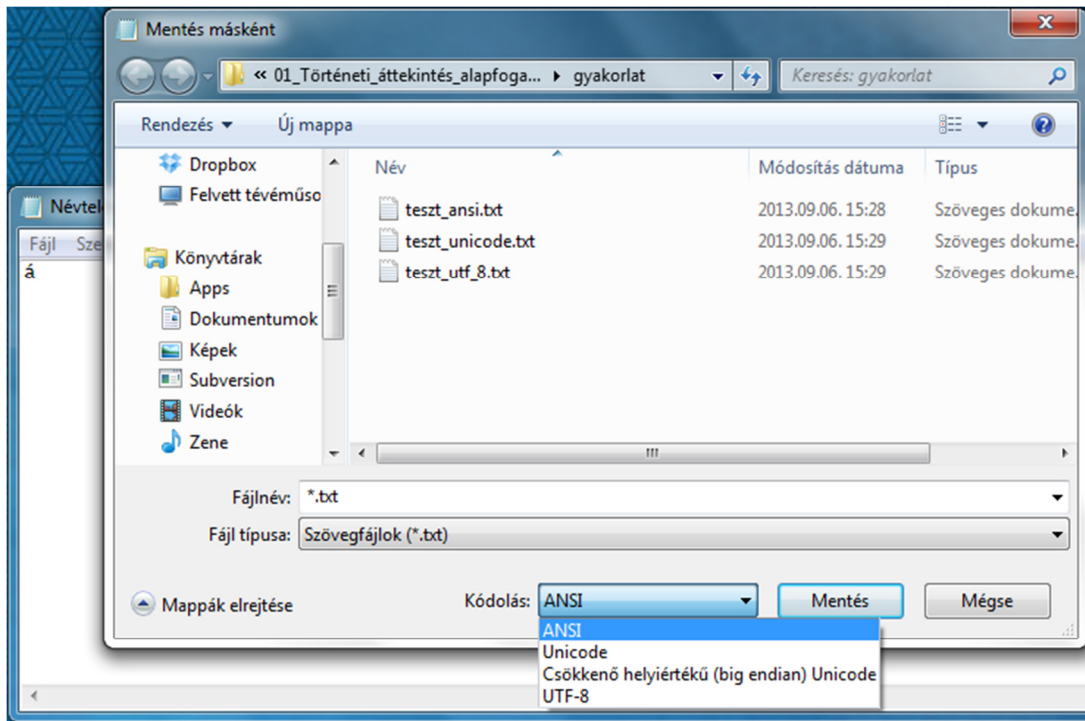
A különböző kódlapok rendszere azonban gyakran okozott bosszúságot idegen nyelvű programok rosszul megjelenő üzeneteinek formájában.

A megoldást egy egységes, minden nemzet minden szimbólumát tartalmazó kódrendszer kidolgozása jelentette illetve jelenti.

Ez az egységes kódrendszer az UNICODE, amely minden egyes szimbólumhoz egy 32 bites számot rendel, vagyis egy karakter tárolása négyszer több helyet foglalna, mint az ASCII esetében. Helytakarékosági szempontból vezették be az UTF-16 és UTF-8 változó hosszúságú kódolási eljárásokat.

Feladat:

Hozzunk létre a windows jegyzetömb alkalmazásával egy teljesen üres szöveges állományt, és mentjük el különböző kódolással (ANSI, UNICODE, utf-8).



Vizsgáljuk meg a létrejött állományokat:

- milyen hosszúak?
- mit tartalmaznak? (használjuk a Total Commander nézőke beállítások hexa funkció-ját)

Az UNICODE és utf-8 kódolású állományok opcionálisan tartalmazhatnak egy bájt sorrend jelzésére alkalmas jelet (BOM Byte Order Mark). Az üres állományokban ezt a kódot (EF EE, FE FF, EF BB BF) láthatjuk.

Feladat:

Ismételjük meg az előző feladatot úgy, hogy egyetlen betűt helyezünk el az állományban

Vizsgáljuk meg ismét a létrejött állományokat:

- milyen hosszúak?
- mit tartalmaznak?

Próbáljuk meg beazonosítani az elhelyezett betű kódját az előadás fóliák és a következő táblázat alapján:

	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F
00	NUL 0000	STX 0001	SOT 0002	ETX 0003	EOT 0004	ENQ 0005	ACK 0006	BEL 0007	BS 0008	HT 0009	LF 000A	VT 000B	FF 000C	CR 000D	SO 000E	SI 000F
10	DLE 0010	DC1 0011	DC2 0012	DC3 0013	DC4 0014	NAK 0015	SYN 0016	ETB 0017	CAN 0018	EM 0019	END 001A	ESC 001B	FS 001C	GS 001D	RS 001E	US 001F
20	SP 0020	! 0021	" 0022	# 0023	\$ 0024	% 0025	& 0026	' 0027	(0028) 0029	* 002A	+ 002B	, 002C	- 002D	. 002E	/ 002F
30	0 0030	1 0031	2 0032	3 0033	4 0034	5 0035	6 0036	7 0037	8 0038	9 0039	: 003A	; 003B	< 003C	= 003D	> 003E	? 003F
40	@ 0040	A 0041	B 0042	C 0043	D 0044	E 0045	F 0046	G 0047	H 0048	I 0049	J 004A	K 004B	L 004C	M 004D	N 004E	O 004F
50	P 0050	Q 0051	R 0052	S 0053	T 0054	U 0055	V 0056	W 0057	X 0058	Y 0059	Z 005A	[005B	\ 005C] 005D	^ 005E	_ 005F
60	` 0060	a 0061	b 0062	c 0063	d 0064	e 0065	f 0066	g 0067	h 0068	i 0069	j 006A	k 006B	l 006C	m 006D	n 006E	o 006F
70	p 0070	q 0071	r 0072	s 0073	t 0074	u 0075	v 0076	w 0077	x 0078	y 0079	z 007A	{ 007B	 007C	} 007D	~ 007E	DEL 007F
80	€ 0080	£ 0081	¥ 0082	₹ 0083	₠ 0084	₡ 0085	₢ 0086	₣ 0087	₤ 0088	₥ 0089	₦ 008A	₧ 008B	₨ 008C	₪ 008D	€ 008E	₭ 008F
90	₮ 0090	₯ 0091	₰ 0092	₱ 0093	₲ 0094	₳ 0095	₴ 0096	₵ 0097	₶ 0098	₷ 0099	₸ 009A	₹ 009B	₺ 009C	₻ 009D	₼ 009E	₽ 009F
A0	₾ 00A0	₿ 00A1	₿ 00A2	₿ 00A3	₿ 00A4	₿ 00A5	₿ 00A6	₿ 00A7	₿ 00A8	₿ 00A9	₿ 00AA	₿ 00AB	₿ 00AC	₿ 00AD	₿ 00AE	₿ 00AF
B0	° 00B0	± 00B1	² 00B2	³ 00B3	´ 00B4	µ 00B5	¶ 00B6	· 00B7	¸ 00B8	¹ 00B9	º 00BA	» 00BB	¼ 00BC	½ 00BD	¾ 00BE	¿ 00BF
C0	À 00C0	Á 00C1	Â 00C2	Ã 00C3	Ä 00C4	Å 00C5	Ç 00C6	È 00C7	É 00C8	Ê 00C9	Ë 00CA	Ì 00CB	Í 00CC	Î 00CD	Ï 00CE	Ð 00CF
D0	Ñ 00D0	Ò 00D1	Ó 00D2	Ô 00D3	Õ 00D4	Ö 00D5	× 00D6	Ø 00D7	Ù 00D8	Ú 00D9	Û 00DA	Ü 00DB	Ý 00DC	Þ 00DD	ß 00DE	À 00DF
E0	à 00E0	á 00E1	â 00E2	ã 00E3	ä 00E4	å 00E5	ç 00E6	è 00E7	é 00E8	ê 00E9	ë 00EA	ì 00EB	í 00EC	î 00ED	ï 00EE	ð 00EF
F0	ñ 00F0	ó 00F1	ô 00F2	õ 00F3	ö 00F4	÷ 00F5	ø 00F6	ù 00F7	ú 00F8	û 00F9	ü 00FA	ý 00FB	þ 00FC	ÿ 00FD	ÿ 00FE	ÿ 00FF

Műveletek logikai értékekkel, Boole algebra.

NEM (NOT)

Operandus (A)	
HAMIS	IGAZ
IGAZ	HAMIS

ÉS (AND)

Operandus (A)	Operandus (B)	Eredmény ($A \wedge B$)
HAMIS	HAMIS	HAMIS
HAMIS	IGAZ	HAMIS
IGAZ	HAMIS	HAMIS
IGAZ	IGAZ	IGAZ

VAGY (OR)

Operandus (A)	Operandus (B)	Eredmény ($A \vee B$)
HAMIS	HAMIS	HAMIS
HAMIS	IGAZ	IGAZ
IGAZ	HAMIS	IGAZ
IGAZ	IGAZ	IGAZ

KIZÁRÓ VAGY (XOR)

Operandus (A)	Operandus (B)	Eredmény ($A \neq B$)
HAMIS	HAMIS	HAMIS
HAMIS	IGAZ	IGAZ
IGAZ	HAMIS	IGAZ
IGAZ	IGAZ	HAMIS

PÉLDÁK

1. A = IGAZ, B = HAMIS, C = ?

$$C = \neg(A \wedge B \vee (B \neq B))$$

a. $C = \neg(IGA Z \wedge HAMIS \vee (HAMIS \neq HAMIS))$

b. $C = \neg(IGA Z \wedge HAMIS \vee HAMIS)$

c. $C = \neg(HAMIS \vee HAMIS)$

d. $C = \neg HAMIS$

e. $C = IGA Z$

2. A = IGAZ, B = HAMIS, C = ?

$$C = \neg(\neg A \vee \neg B) \neq \neg(\neg A \wedge \neg B)$$

a. $C = \neg(\neg IGA Z \vee \neg HAMIS) \neq \neg(\neg IGA Z \wedge \neg HAMIS)$

b. $C = \neg(HAMIS \vee IGA Z) \neq \neg(HAMIS \wedge IGA Z)$

c. $C = \neg(IGA Z) \neq \neg(HAMIS)$

d. $C = HAMIS \neq IGA Z$

e. $C = IGA Z$

3. A = IGAZ, B = HAMIS, C = IGAZ, D = ?

$$D = \neg C \neq A \vee B \vee (A \wedge \neg A) \neq (C \neq B)$$

a. $D = \neg IGA Z \neq IGA Z \vee HAMIS \vee (IGA Z \wedge \neg IGA Z) \neq (IGA Z \neq HAMIS)$

b. $D = HAMIS \neq IGA Z \vee HAMIS \vee HAMIS \neq IGA Z$

c. $D = HAMIS$

ÖNÁLLÓ FELADAT

1. A = HAMIS, B = HAMIS, C = IGAZ, D = ?

$$D = \neg (A \vee A) \neq (B \vee B) \neq (C \wedge C)$$

2. A = HAMIS, B = IGAZ, C = IGAZ, D = ?

$$D = \neg (A \vee (B \wedge (C \neq B)) \vee C) \neq (C \wedge C)$$

3. A = HAMIS, B = HAMIS, C = HAMIS, D = ?

$$D = C \vee \neg (A \vee A \wedge C) \neq (C \wedge \neg C) \wedge \neg B$$

