Tartalom

A gyakorlósor lektorálatlan, hibákat tartalmazhat, és minden bizonnyal tartalmaz is. Használata csak kellő körültekintéssel javasolt! A felfedezett hibákat a szeghalmy.szilvia@inf.unideb.hu címen lehet jelezni.

1. Számrendszerek közti átváltás

Tízesből kettes számrendszerbe

1) 100 = $_2$
2) 140 = $_2$
3) 250 = $_2$
4) 120.025 = $_2$
5) 280.4 = $_2$

Tízesből nyolcas számrendszerbe

1) 100 = $_8$
2) 140 = $_8$
3) 250 = $_8$
4) 120.025 = $_8$
5) 280.4 = $_8$

Kettesből tizenhatos számrendszerbe

1) 10010101 = $_{16}$
2) 10110010101 = $_{16}$
3) 11010010.0110010 = $_{16}$
4) $11111010.\overline{0110}_2$ = $_{16}$
5) $111110101.10\overline{101}_2$ = $_{16}$

Kettesből nyolcas számrendszerbe

1) 10010101 = $_8$
2) 10110010101 = $_8$
3) 11010010.0110010 = $_8$
4) $11111010.\overline{0110}_2$ = $_8$
5) $111110101.10\overline{101}_2$ = $_8$

Kettesből tízes számrendszerbe

1) 1101 = $_{10}$
2) 10010101 = $_{10}$
3) 11010.011 = $_{10}$
4) 1011.01 = $_{10}$
5) $110.\overline{1}_2$ = $_{10}$

Nyolcasból kettes számrendszerbe

1) 100_8 = $_2$
2) 140_8 = $_2$
3) 250_8 = $_2$
4) 120.025_8 = $_2$
5) 280.4_8 = $_2$

Nyolcasból tízes számrendszerbe

1) 100_8 = $_{10}$
2) 140_8 = $_{10}$
3) 120.025_8 = $_{10}$
4) 103.4_8 = $_{10}$

Tizenhatosból kettes számrendszerbe

1) $1B_{16}$ = $_2$
2) $1E0_{16}$ = $_2$
3) 25_{16} = $_2$
4) $92.2A_8$ = $_2$

Tizenhatosból tízes számrendszerbe

1) $9E_{16}$ = $_{10}$
2) $1E0_{16}$ = $_{10}$
3) 25_{16} = $_{10}$
4) $92.2A_8$ = $_{10}$

1.1. Megoldások

Tízesből kettes számrendszerbe

1) $100 = 1100100_2$

100	:2
50	0
25	0
12	1
6	0
3	0
1	1
0	1

2) $140 = 10001100_2$
3) $250 = 11111010_2$
4) $120.025 = 1111000.000\overline{00011}_2$

120	:2
60	0
30	0
15	0
7	1
3	1
1	1
0	1

0	025 *2
0	05
0	1
0	2
0	4
0	8
1	6
1	2

5) $280.4 = 100011000.\overline{0110}_2$

Tízesből nyolcas számrendszerbe

Tipp: Aki nem szeret nyolccal osztani, váltsa át a számot kettes számrendszerbe, majd a kapott értéket váltsa tovább nyolcas számrendszerbe.

$100 = 144_8$

100	:8
12	4
1	4
0	1

1) $140 = 214_8$
2) $250 = 372_8$

3) $120.025 = 170.01\overline{463}_8$

120	:8
15	0
1	7
0	1

0	025 *8
0	2
1	6
4	8
6	4
3	2

4) $280.4 = 430.\overline{3146}_8$

Kettesből tizenhatos számrendszerbe

A megoldás során négyesével tagoljuk a törtponttól kiindulva a számjegyeket mindkét irányban. Amennyiben a legelső vagy a legutolsó blokknál nincs meg a négy számjegy, egészítsük ki 0-val a „blokkot”. Az így kapott négy bináris számjegyet tartalmazó blokkok pontosan egy hexadecimális számjegy kettes számrendszerbeli megfelelői. (pl: $1100_2 = 12_{10} = C_{16}$)

1) 1001 0101 = 95_{16}, mert az $1001_2 = 9$ és a $0101_2 = 5$
2) *0*101 1001 0101 = 595_{16}
3) 1101 0010.0110 010*0* = $D2.64_{16}$
4) $1111\ 1010.\overline{0110}_2$ = $FA.\overline{6}_{16}$
5) $1\ 1111\ 0101.10\overline{101}_2$ = 0001 1111 0101. 1010 1101 1011 0110… =$1F5.ADB6\ \ldots_{16}$

Kettesből nyolcas számrendszerbe

A megoldás a tizenhatos számrendszernél tanulthoz hasonló, azonban most hármasával tagoljuk a számjegyeket. (Természetesen most is a törtponttól indulunk ki, bal és jobb oldali irányban is.)

1) *0*10 010 101 = 225_8, mert a $010_2 = 2$, a $010_2 = 2$ és az $101_2 = 5$
2) *0*10 110 010 101 = 2625_8
3) *0*11 010 010.011 001 = 322.31_8
4) $11\ 111\ 010.\overline{0110}_2$ = 011 111 010.011 001 100 110 = 372.3146_8
5) $111\ 110\ 101.10\overline{101}_2$ = 111 110 101.101 011 011 011… = $765.533\ldots_8 = 765.5\overline{3}_8$

Kettesből tízes számrendszerbe

1) $1101 = 1*2^3 + 1*2^2 + 0*2^1 + 1*2^0 = 8+4+0+1 = 13$
2) 10010101 = 149
3) $11010.011 = 26+ 0*2^{-1} + 1*2^{-2} + 1*2^{-3} = 26+0.25+0.125 = 26.375$
4) 1011.01 = 11.25
5) $110.\overline{1}_2 = 6 + 1*2^{-1} + 1*2^{-2} + 1*2^{-3}\ldots = 6+\sum_{i=1}^{\infty}\frac{1}{2^i} \approx 7$

Nyolcasból kettes számrendszerbe
Minden nyolcas számrendszerbeli számjegy leírható három biten. Az átváltás során egyszerűen minden számjegyhez rendeljük hozzá a neki megfelelő bináris alakot 3 biten(!). Fontos, hogy akkor is

három bitet rendeljünk egy számjegyhez, ha az érték kisebb bitszámon is elférne. Pl. a 0 nyolcas számrendszerbeli számjegyhez 000-át rendelünk. Természetesen a szám elejéről és a törtpont után a szám végéről a 0-ák szokás szerint elhagyhatók.

1) 100_8 = 001 000 000 = $1\ 000\ 000_2$
2) 140_8 = 001 100 000 = $1\ 100\ 000_2$
3) 250_8 = 010 101 000 = $10\ 101\ 000_2$
4) 120.025_8 = 001 010 000.000 010 101 = $1\ 010\ 000.000\ 010\ 101_2$
5) 280.4_8 = Csak gonoszkodtam. Nincs ilyen nyolcas számrendszerbeli szám. :)

Tizenhatosból kettes számrendszerbe

A stratégia az előzővel analóg, azonban minden számjegyhez 4 bitet fogunk rendelni, hiszen ennyi bit szükséges a legnagyobb 16-os számrendszerbeli számjegy (F_{16}) 2-es számrendszerbeli tárolásához (1111_2).

1) $1B_{16}$ = 0001 1011 = $1\ 1011_2$
2) $1E0_{16}$ = 0001 1110 0000 = $1\ 1110\ 0000_2$
3) 25_{16} = 0010 0101 = $10\ 0101_2$
4) $92.2A_8$ = 1001 0010.0010 1010_2

Nyolcasból tízes számrendszerbe

1) $100_8 = 1*8^2 + 0*8^1 + 0*8^0 = 64_{10}$
2) $140_8 = 96_{10}$
3) $120.025_8 = 80 + 0*8^{-1} + 2*8^{-2} + 5*8^{-3} = 80 + \frac{2}{64} + \frac{5}{512} = 80\frac{21}{512} = 80\frac{3}{73}$
4) $103.4_8 = 67.5_{10}$

Tizenhatosból tízes számrendszerbe

1) $9E_{16} = 9*16^1 + E*16^0 = 9*16 + 14*1 = 158_{10}$
2) $1E0_{16} = 480_{10}$
3) $25_{16} = 37_{10}$
4) $92.2A_8 = 146 + 2*16^{-1} + A*16^{-2} = 146 + \frac{2}{16} + \frac{10}{256} = 146\frac{21}{128}$

2. Műveletek (+, -, bitműveletek)

Összeadás és kivonás 2-es, 8-as, 16-os számrendszerekben. Végezzük el mind a két műveletet az alábbi számpárokra.

1) $111001_{(2)}$ $1011011_{(2)}$
2) $773.4_{(8)}$ $214.2_{(8)}$
3) $FA24_{(16)}$ $13B3_{(16)}$

Végezze el az összeadás és a kivonás műveletet az alábbi bitsorozatokon, 8 biten!

1) $11101001_{(2)}$ $10010011_{(2)}$
2) $10101001_{(2)}$ $00110011_{(2)}$
3) $10000000_{(2)}$ $00000011_{(2)}$
4) $01111111_{(2)}$ $01010011_{(2)}$

Logikai műveletek

Lásd mat. log. jegyzeted.

Bitműveletek:

~	bitenkénti negált
|	bitenkénti vagy
&	bitenként és
^	bitenkénti kizáró vagy
<<	bitenkénti eltolás balra
>>	bitenkénti eltolás jobbra

Minden feladatnál **1 bájton** dolgozzon. (A bitsorozatoknál a számrendszer jelzését elhagytuk.)

1) ~00111001
2) 00111001 | 10010010
3) 00111001 & 10010010
4) 00111001 ^ 10010010
5) 10110000 << $1_{(10}$
6) 10110001 << $4_{(10}$
7) 10110000 >> $1_{(10}$
8) 10110001 >> $4_{(10}$
9) 11100100 & (~10110011)
10) (11100010 << $2_{(10}$) & 10110011
11) 0xA1 << $3_{(10}$ (A1 hexadecimális érték eltolása 3 bittel balra)
12) Szorozzuk meg a 3-as számot 8-al, bitművelet segítségével.

2.1. Megoldások

Összeadás és kivonás 2-es, 8-as, 16-os számrendszerekben.

A műveletek megvalósítási elve minden számrendszerben megegyezik a tízes számrendszerben használtakkal, csupán az alapszám más. Például kettes számrendszerben a 0+0 = 0, 1+0 = 1, 0+1 = 1, 1+1 = 0 és lesz 1 átvitel, mert csak az alapszámnál kisebb számjegyeink vannak, ha azt túllépjük, az alap levonásra kerül és azt a következő számjegyek összeadásánál vesszük figyelembe.

Ha kivonásnál a kivonandó szám a nagyobb, akkor cseréljük meg a két számot, végezzük el úgy a kivonást, és persze ne feledkezzünk meg a negatív előjel kiírásáról.

A könnyebb követhetőség érdekében a feladatok megoldásánál egy külön C-vel bevezetett sor fogja jelezni az átvitelt. Az átvitelt a keletkezés helyétől egy jeggyel balra csúsztatva írjuk le, hiszen ott kerül majd beszámításba.

1) $111001_{(2)}$ $1011011_{(2)}$

```
    111001
+  1011011
C 1111 11
  10010100    (eredmény)

   1011011 (A kisebb számból kellene kivonni a nagyobbat-> felcseréljük)
-   111001
C  1
   0100010 = -100010 (kisebb szám - nagyobb szám -> negatív az érték)
```

2) $773.4_{(8)}$ $214.2_{(8)}$

```
    773.4   7+1->8, alapszámot levonjuk (8-8 = 0) és lesz átvitel
+   214.2   7+2+1->10, alapszámot levonjuk (10-8=2) és lesz átvitel
C  11
   1207.6

    773.4    3-4 -> alapszám+3 -4 és egy átvitel = 11-4 = 7 + átvitel
-   214.2
C    1
    557.2
```

3) $FA24_{(16)}$ $13B3_{(16)}$

```
    FA24    B+2 = 11+2 = 13 = D
+   13B3    A+3 = 10+3 = 13 = D
C  1        F+1 = 15+1 = 16 - alapszám = 0 és lesz egy átvitel
   10DD7

    FA24   2-B =2-11= 2+alap-11 és átvitel = 18-11 = 7 és átvitel
-   13B3   A-3-1 = 10-4 = 6
C    1     F-1 = 15-1 = 14
    E671
```

Végezze el az összeadás és a kivonás műveletet az alábbi bitsorozatokon! 8 bitet használjon!

1) $11101001_{(2)}$ $10010011_{(2)}$

```
   11101001
+  10010011
C 1     11
   01111100
```

Megj: A feladat szerint 8 bitet kell használnunk, ezért az utolsó átvitelt nem jegyezzük le. Az összeadás során túlcsordulás történt.)

```
   11101001
-  10010011
C    1 11
   01010110
```

2) $10101001_{(2)}$ $00110011_{(2)}$

```
   10101001
+  00110011
C   1   11
   11011100
```

```
   10101001
-  00110011
C  111 11
   01110110
```

3) $10000000_{(2)}$ $00000011_{(2)}$

```
   10000000
+  00000011
C
   10000011
```

```
   10000000
-  00000011
C  1111111
   01111101
```

4) $01111111_{(2)}$ $01010011_{(2)}$

```
   01111111
+  01010011
C  1111111
   11010010
```

```
   01111111
-  01010011
C
   00101100
```

Bitműveletek megoldásai

1) ~00111001 = 11000110, mert a bitenkénti negálás során a 0 értékek 1-re, az 1-es értékű bitek 0-ra változnak.

2) 00111001 | 10010010 = 10111011, mert egy eredmény bit akkor és csak akkor 0, ha a *vagy*olandó bitsorozatok azonos pozíción lévő bitjei közül mindkettő 0 értékű.

```
  00111001 (egymás alá írtuk a két sorozatot, így könnyen látható
| 10010010   mely bitek helyezkednek el azonos pozíción)
  10111011 (eredmény)
```

3) 00111001 & 10010010 = 00010000, mert egy eredménybit akkor és csak akkor 1, ha az *és*elendő sorozat azonos pozícióján lévő bitek közül mindkét bit 1-es értékű.

```
  00111001
& 10010010
  00010000 (eredmény)
```

4) 00111001 ^ 10010010 = 10101011, mert egy eredménybit értéke akkor és csak akkor egy, ha a *kizáró vagy*ozandó bitsorozatok azonos pozíción lévő bitjei közül az egyik, és csak az egyik 1-es értékű. (Tehát a *vagy*tól abban különbözik, hogy az 1 ^ 1 értéke is 0-át eredményez).

```
  00111001
^ 10010010
  10101011 (eredmény)
```

5) 10110000 << 1 = 01100000***0***, mert az egy bittel balra való *eltolás* műveletnél minden bit egy pozícióval kerül balra. A legmagasabb helyiértékű bit ennek hatására eltűnik (pirossal jelölve), a legalacsonyabb helyiértékű bitre pedig 0 kerül (dőlt, félkövérrel jelölve).

6) 10110001 << 4 = 0001***0000***, mert több bittel való eltolás művelet pontosan ugyanazt az eredményt adja, mint ha az 1 bittel való eltolást négyszer egymás után végrehajtanánk a bitsorozatra.

7) 10110000 >> 1 = ***0***1011000, mert az egy bittel jobbra való eltolás műveletnél minden bit egy pozícióval kerül jobbra. A legkisebb helyiértékű bit ennek hatására eltűnik (pirossal jelölve), a legmagasabb helyiértékű bitre pedig 0 érték kerül (dőlt, félkövér).

8) 10110001 >> 4 = ***0000***1011, mert több bittel való eltolás művelet pontosan ugyanazt az eredményt adja, mint ha az 1 bittel való eltolást négyszer egymás után végrehajtanánk a bitsorozatra.

9) 11100100 & (~10110011)
Végezzük el a bitenkénti negálást 10110011 sorozatra: 01001100

```
  11100100 (egy eredmény bit 1 <=> ha az éselendő mindkét bit 1 volt)
& 01001100
  01000100
```

10) (11100010 << 2) & 10110011
Az eltolás eredménye 1110001*00* lenne, viszont a feladat szerint 8 biten dolgozunk, ezért a legfelső két bit az eltolás hatására elveszik (túlcsordul). 10001000 bitsorozatot adva.

```
  10001000 (egy eredmény bit 1 <=> ha a éselendő mindkét bit 1 volt)
& 10110011
  10000000
```

11) 0xA1 << 3 = 1010 $0001_{(2}$ << 3 = 0000 1***000***

12) Szorozzuk meg a 3-as számot 8-cal, bitművelet segítségével.

Használjuk ki, hogy egy bináris szám végére egy nulla értéket fűzve a szám értéke a duplájára nő. Pl.: $11_{(2)}$, $110_{(2)}$, $1100_{(2)}$ decimális értékei rendre: 3, 6, 12. Tehát egy szám 2^n-nel való szorzása megegyezik az *x* << *n* alakkal.
A fentiek alapján, a feladat megoldásához a hármas szám bináris számrendszerbeli alakját három bittel kell balra tolni, hiszen $2^3 = 8$, ezért az *n* értéke éppen három.

Lássuk rendben van-e az eredmény. A 3-ast váltsuk át kettes számrendszerbe, hogy bitenként lássuk. Végezzük el az eltolást.

$11_{(2)}$ << 3 = $11000_{(2)}$

És váltsuk vissza az eredményt 10-es számrendszerbe:

$1*2^4+1*2^3 = 16+8 = 24$, ez valóban $3*8$, tehát a megoldás helyes.

Megjegyzés: A programozás során észben kell tartani a szám reprezentációját. Ha már az adatábrázoláson is túl vagy, akkor térj vissza ide és gondold át, mely reprezentációk esetében használható ez a trükk a kettő hatvánnyal való szorzásra.

3. Számítógépes adatábrázolás

Előjel nélküli egészszámok ábrázolása

1) 126 (1 bájton)
2) 211 (1 bájton)
3) 30 (1 bájton)
4) 30.45 (1 bájton)
5) 0000 0011 (Mi volt az eredeti szám, ha ez a reprezentációja?)
6) 1100 0001 (Mi volt az eredeti szám, ha ez a reprezentációja?)

Előjeles egész számok ábrázolása - előjelbittel

1) -126 (1 bájton)
2) 211 (2 bájton)
3) 30 (1 bájton)
4) -30 (1 bájton)
5) 0000 0011 (Mi volt az eredeti szám, ha ez a reprezentációja?)
6) 1100 0001 (Mi volt az eredeti szám, ha ez a reprezentációja?)

Előjeles egész számok ábrázolása - eltolással

1) -126 (1 bájton, 128-többlettel)
2) 211 (2 bájton, 2^{15} többlettel)
3) 20 (6 biten, 32 többlettel)
4) -20 (6 biten, 32 többlettel)
5) 0000 0011 (Mi volt az eredeti szám, ha ez a reprezentációja? (128-többlettel))
6) 1100 0001 (Mi volt az eredeti szám, ha ez a reprezentációja? (128-többlettel))

Előjeles egész számok ábrázolása – negatív számok kettes komplemenskóddal

1) -126 (1 bájton)
2) 211 (2 bájton)
3) -1 (1 bájton)
4) -20 (1 bájton)
5) 0000 0011 (Mi volt az eredeti szám, ha ez a reprezentációja?)
6) 1100 0001 (Mi volt az eredeti szám, ha ez a reprezentációja?)

Lebegőpontos számábrázolás

1) -126.25 (2 bájton, rejtett 1-essel, 6 bites eltolt (2^5-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
2) 211.4 (4 bájton, rejtett egyessel, 7 bites eltolt (2^6-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
3) -1 (2 bájton, rejtett egyessel, 7 bites eltolt (2^6-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)

4) 0.2 (2 bájton, rejtett egyessel, 6 bites eltolt (2^5-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
5) 0.2 (2 bájton, **rejtett egyes nélkül,** 6 bites eltolt (2^5-1) karakterisztikával, kettes alapra normáltan.
6) Mi volt az eredeti, 10-es számrendszerbeli értéke az alábbi sorozatnak, ha lebegőpontos ábrázolást használtunk 2 bájton, 8 bites karakterisztikával (2^7-1) eltolás, kettes alapra normáltan, rejtett 1-essel. A reprezentáció hexadecimális formában: C3A0

3.1. Megoldások

Előjel nélküli egészszámok ábrázolása (egyeneskód)

Egész számok ábrázolásáról van szó, ezért csak a szám egészrészével foglalkozunk. Váltsuk át kettes számrendszerbe. Ha szükséges, akkor balról egészítsük ki a számot a megfelelő bitszámra.

1) 126 (1 bájton) $\rightarrow$ 111 $1110_2 \rightarrow$ 0111 1110
2) 211 (1 bájton) $\rightarrow$ $11010011_2 \rightarrow$ 1101 0011
3) 30 (1 bájton) $\rightarrow$ 11110 $\rightarrow$ 0001 1110
4) 30.45 (1 bájton) $\rightarrow$ csak a 30 kerül tárolásra, mint az előbb, hiszen egész számot ábrázolunk
5) 0000 0011 $\rightarrow$ $1*2^1+1*2^0 = 3$
6) 1100 0001 $\rightarrow$ $1*2^7+1*2^6+1*2^0 = 193$

Előjeles egész számok ábrázolása – előjelbittel (félkövérrel)

Az előjel 0, ha a szám pozitív, 1, ha a szám negatív. A reprezentáció első bitje (legfelső helyérték) ez lesz. A maradék bitszámon a szám abszolútértékét, mint előjelnélküli egész számot ábrázoljuk.

1) -126 (1 bájton) $\rightarrow$ -111 $1110_2 \rightarrow$ **1**111 1110
2) 211 (2 bájton) $\rightarrow$ $11010011_2 \rightarrow$ **0**000 0000 1101 0011 (egy bájton nem férne ki!)
3) 30 (1 bájton) $\rightarrow$ 11110 $\rightarrow$ **0**001 1110
4) -30 (1 bájton) $\rightarrow$ -11110 $\rightarrow$ **1**001 1110
5) 0000 0011 $\rightarrow$ $1*2^1+1*2^0 = 3$
6) **1**100 0001 $\rightarrow$ $-1*2^6+1*2^0 = -65$

Előjeles egész számok ábrázolása – eltolással

Az ábrázolandó számhoz adjuk hozzá a tároláshoz használt többletet, majd ezt az értéket, mint előjel nélküli egészt ábrázoljuk.

1) -126 (1 bájton, 128-többlettel) $\rightarrow$ -126 +128 = 2 $\rightarrow$ $10_2 \rightarrow$ 00000010
2) 211 (2 bájton, 2^{15} többlettel) $\rightarrow$ 0000 0000 1101 0011 + 1000 0000 0000 0000
   $\rightarrow$ 1000 0000 1101 0011
3) 20 (6 biten, 32 többlettel) $\rightarrow$ 20+32 = 52 = $\rightarrow$ 110100

4) -20 (6 biten, 32 többlettel) $\rightarrow$ -20+32 = 12 = $\rightarrow$ 001100
5) 0000 0011 $\rightarrow$ $1*2^1+1*2^0 = 3 - 2^7 = -125$
6) 1100 0001 $\rightarrow$ $1*2^7+1*2^6+1*2^0 - 2^7 = 65$

Előjeles egész számok ábrázolása – negatív számok kettes komplemenskóddal

1. Írjuk fel a számot kettes számrendszerben, és balról egészítsük ki 0 értékekkel (egyeneskód).
2. Amennyiben a szám negatív, akkor
   a. Képezzük az egyes komplemenst a bitek negálásával.
   b. Adjuk hozzá +1-et az egyes komplemenshez.

Tehát csak a negatív számoknál képezzük a kettes komplemenst, a pozitív számoknál csak az 1-es lépést alkalmazzuk.

Amennyiben a reprezentációból kell az eredeti értéket meghatároznunk, akkor a legfelső bit ismeretében dönthetünk arról, hogy a szám pozitív-e (0) vagy negatív (1). Előbbi esetben csak fel kell írni a megfelelő decimális értéket, utóbbi esetén előbb visszaalakítjuk egyeneskódba a reprezentációt (-1, majd negálás).

1) -126 (1 bájton) $\rightarrow$ -111 1110_2
   $\rightarrow$ 0111 1110 (kiegészítve)
   $\rightarrow$ 1000 0001 (bitek negálva)
   $\rightarrow$ 1000 0010 (+1 hozzáadása után)

2) 211 (2 bájton) $\rightarrow$ 1101 0011_2
   $\rightarrow$0000 0000 1101 0011 (pozitív érték, ezért nem képezzük a komplemenst)

3) -1 (1 bájton) $\rightarrow$ 1_2
   $\rightarrow$ 0000 0001 (kiegészítve)
   $\rightarrow$ 1111 1110 (bitek negálva)
   $\rightarrow$ 1111 1111 (+1 hozzáadása után)

4) -20 (1 bájton) $\rightarrow$ 10100_2 $\rightarrow$
   $\rightarrow$ 0001 0100
   $\rightarrow$ 1110 1011 (bitek negálva)
   $\rightarrow$ 1110 1100 (+1 hozzáadása után)

5) 0000 0011 = 3 (mivel nulla a legfelső bit, ezért a szám pozitív előjeles egészként tekinthető)

6) 1100 0001 (mivel a legfelső bit 1, ezért a szám negatív, és kettes komplemensben van)
   $\rightarrow$ 11000000 (-1 után)
   $\rightarrow$ 00111111 (bitek negálása után)
   $\rightarrow$ - $(1*2^5+1*2^4+1*2^3+1*2^2+1*2^1+1*2^0) = -63$

Lebegőpontos számábrázolás

Jelölések: **előjelbit**, eltolt karakterisztika, *kitöltő nulla*

1) -126.25 (2 bájton, rejtett 1-essel, 6 bites eltolt (2^5-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
   kettes számrendszerbeli alak: -1111110.01_2
   normált alak: -1.11 1110 01 * 2^6 (tehát a karakterisztika 6, a mantissza: -1.11...)
   eltolt karakterisztika: 6 + 2^5-1= 37 = 100101_2
   reprezentáció: előjel, eltolt karakterisztika (6 biten), mantissza, a rejtett 1-es miatt az egészrészt elhagyjuk. Azért tehető ez meg, mert a normalizálás miatt a 0 kivételével mindig 1-es számjegy áll a szám elején, a nullát pedig speciális csupa nulla sorozat jelzi.)
   reprezentáció: **1**100 1011 1111 0010

reprezentáció hexadecimális alakban: CBF2

2) 211.4 (4 bájton, rejtett 1-essel, 7 bites eltolt (2^6-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
kettes számrendszerbeli alak: $11010011.\overline{0110}$ (ismétlődő)
normált alak: 1.101 0011 0110... *2^7 (tehát a karakterisztika 7, a mantissza: 1.101...)
eltolt karakterisztika: 7 + 63 = 1000110_2
reprezentáció: **0**100 0110 1010 0110 1100 1100 1100 1100
reprezentáció hexadecimális alakban: 46A6CCCC

3) -1 (2 bájton, rejtett 1-essel, 7 bites eltolt (2^6-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
kettes számrendszerbeli alak: -1
normált alak: -1 *2^0 (tehát a karakterisztika 0, a mantissza: -1.0, de az 1-est elrejtjük)
eltolt karakterisztika: 0 + 63 = *0*111111_2 *(kiegészítve 7 bitre)*
reprezentáció: **1**011 1111 0000 0000
reprezentáció hexadecimális alakban: BF00

4) 0.2 (2 bájton, rejtett egyessel 6 bites eltolt (2^5-1) karakterisztikával, kettes alapra normáltan. A bájtok tartalmát hexadecimálisan adjuk meg.)
kettes számrendszerbeli alak: $0.\overline{0011}$ = 0.0011 0011...
normált alak: 0001.1 0011... = 1.1 0011.. *2^{-3} (tehát a karakterisztika -3, a mantissza: 1.1001100110011...)
eltolt karakterisztika: -3 + 31 = 11100_2 = *0*11100 (kiegészítve 6 bitre)
reprezentáció: **0**011 1001 0011 0011
reprezentáció hexadecimális alakban: 3933

5) 0.2 (2 bájton, **rejtett egyes nélkül**, 6 bites eltolt (2^5-1) karakterisztikával, kettes alapra normáltan.
Változás a 4. feladathoz képest csak a reprezentáció felírásánál adódik. A mantissza (1.1001100110011…) törtponttól balra eső 1-es értékét is szerepeltetjük a sorozatban
reprezentáció: **0**011 1001 1001 1001
reprezentáció hexadecimális alakban: 3999

6) Mi volt az eredeti, 10-es számrendszerbeli értéke az alábbi sorozatnak, ha lebegőpontos ábrázolást használtunk 2 bájton, 8 bites karakterisztikával (2^7-1) eltolás, kettes alapra normáltan, rejtett 1-essel. A reprezentáció hexadecimális formában: C3A0
A reprezentáció bitjei: 1100 0011 1010 0000
Bontsuk szét előjelre, karakterisztikára, mantisszára:

1	100 00111	**1.**010 0000
előjel	eltolt karakterisztika	mantissza (a rejtett 1-est visszahoztuk)

Az eltolt karakterisztika decimális értéke 135 2^7-1.
A karakterisztika értéke eltolás nélkül: 135 – (2^7-1) = 135 - 127 = 8
Ebből már felírható a 2-es alapra normalizált alak: - $1.010\ 0000_{(2}$ * 2^8

A kettes számrendszerbeli alak: $-101000000_{(2}$
10-es számrendszerbe visszaváltva az értéket: -320