

(a) $01101011 + 10111101$

(b) $01101011 + 11110000$

(c) $10010110 + 10111011$

3 Représentation des réels et des caractères

Exercice 6. Représentation d'une partie fractionnaire

1. Coder sur 8 bits les parties fractionnaires suivantes :

(a) $(0.578125)_{10}$

(b) $(0.85)_{10}$

(c) $(0.11)_{10}$

2. Décoder

(a) $(0.10110000)_2$

(b) $(0.11011001)_2$

(c) $(0.00110101)_2$

Exercice 7. Codage en norme IEEE 754 (simple précision)

Les flottants sont codés sur 32 bits :

- Un bit de signe : 0 pour + et 1 pour -1
- 8 bits pour l'exposant ... décallé : si e est l'exposant du flottant, on retiendra $e + 127$
- 23 bits pour la mantisse : la mantisse étant de la forme $1.c_1c_2\dots$ sur 24 bits, on ne retient que les 23 bits après la virgule

Par exemple considérons le nombre écrit en base 10 : $-17,25$. On écrit ce nombre en base 2 : $(-17,25)_{10} = (-10001,01)_2 = -1.000101_2 \times 2^4$. On trouve alors :

- signe - bit : 1
 - exposant $4 + \text{biais} = 4 + 127 = 131 = 10000011_2$ sur 8 bits.
 - mantisse On retient 0000101_2 (on omet le 1 devant la virgule). Donc sur 23 bits on a donc : 000 0101 0000 0000 0000 000
- Ainsi $(-17,25)_{10}$ est représenté en norme IEEE 754 en simple précision $1100\ 0001\ 1000\ 0101\ 0000\ 0000\ 0000\ 0000$.
Pour simplifier l'écriture (et la lecture), on pourra écrire cette expression en base 16 sous la forme $C1850000_{16}$

1. Coder en IEEE 754 les réels

(a) 40

(b) -0.078125

(c) 13.625

(d) -87.375

2. Décoder les réels exprimés en norme IEEE 754 donnés en simple précision (et en base 16) :

(a) $41FE8000_{16}$

(b) $3EA80000_{16}$

(c) $C5E00000_{16}$

(d) 00380000_{16}

Exercice 8. Code ASCII : représentation des caractères

ASCII								
(.) ₁₀	(.) ₁₆	char	(.) ₁₀	(.) ₁₆	char	(.) ₁₀	(.) ₁₆	char
32	0x20	<SPACE>	64	0x40	@	96	0x60	`
33	0x21	!	65	0x41	A	97	0x61	a
34	0x22	"	66	0x42	B	98	0x62	b
35	0x23	#	67	0x43	C	99	0x63	c
36	0x24	\$	68	0x44	D	100	0x64	d
37	0x25	%	69	0x45	E	101	0x65	e
38	0x26	&	70	0x46	F	102	0x66	f
39	0x27	'	71	0x47	G	103	0x67	g
40	0x28	(72	0x48	H	104	0x68	h
41	0x29)	73	0x49	I	105	0x69	i
42	0x2A	*	74	0x4A	J	106	0x6A	j
43	0x2B	+	75	0x4B	K	107	0x6B	k
44	0x2C	,	76	0x4C	L	108	0x6C	l
45	0x2D	-	77	0x4D	M	109	0x6D	m
46	0x2E	.	78	0x4E	N	110	0x6E	n
47	0x2F	/	79	0x4F	O	111	0x6F	o
48	0x30	0	80	0x50	P	112	0x70	p
49	0x31	1	81	0x51	Q	113	0x71	q
50	0x32	2	82	0x52	R	114	0x72	r
51	0x33	3	83	0x53	S	115	0x73	s
52	0x34	4	84	0x54	T	116	0x74	t
53	0x35	5	85	0x55	U	117	0x75	u
54	0x36	6	86	0x56	V	118	0x76	v
55	0x37	7	87	0x57	W	119	0x77	w
56	0x38	8	88	0x58	X	120	0x78	x
57	0x39	9	89	0x59	Y	121	0x79	y
58	0x3A	:	90	0x5A	Z	122	0x7A	z
59	0x3B	;	91	0x5B	[123	0x7B	}
60	0x3C	<	92	0x5C	\	124	0x7C	
61	0x3D	=	93	0x5D]	125	0x7D	{
62	0x3E	>	94	0x5E	^	126	0x7E	~
63	0x3F	?	95	0x5F	_	127	0x7F	

1. En utilisant le code ASCII, écrire vos noms et prénoms sans oublier de mettre les initiales en majuscules (on oubliera les accents et cédilles)
2. En utilisant le code ASCII, décoder la phrase suivante : 76 39 97 115 99 105 105 44 32 99 39 101 115 116 32 102 97 99 105 108 101 46
3. En utilisant le code ASCII, décoder la phrase hexadécimal : 4A 27 41 49 20 54 52 4F 55 56 45 20 21