

## 1 Travaux Dirigés

**Exercice 1.** Écrire une fonction `int quotient(int a, int b)` qui calcule et retourne le quotient de la division euclidienne de  $a$  par  $b$  (le quotient est obtenu par soustractions successives). On écrira cette fonction à l'aide d'une boucle `while`.

**Exercice 2.** On considère l'algorithme suivant calculant la partie entière du logarithme en base 2 d'un entier  $n$ , c'est à dire l'entier  $l2(n)$  tel que  $2^{l2(n)} \leq n < 2^{l2(n)+1}$ .

```
l2 = 0;
p2 = 2; // contient 2^(l2+1)
tant que p2 <= n faire
    p2 = 2 * p2;
    l2 = l2 + 1;
fintant que;
retourne(l2);
```

Exécuter cet algorithme pour  $n = 7$ .

En vous inspirant de cet algorithme, écrire une fonction `int l2(long n)`.

CARACTÈRES

**Exercice 3.** Étudiez la page du manuel en ligne fournie en annexe.

Lire et comprendre le programme suivant :

```
#include <stdlib.h>
#include <stdio.h>

int
main (void)
{
    for (char c = '\0'; c < '~'; c++)
        printf ("%3d \t %c\n", c, c);
    return EXIT_SUCCESS;
}
```

**Exercice 4.** Le but de cet exercice est d'écrire un ensemble de fonctions sur les caractères similaires à celles que l'on peut trouver dans la bibliothèque standard `libc`. Contrairement aux fonctions de la `libc`, on utilisera ici le type `bool`.

Écrire le programme `caractere` contenant les fonctions suivantes :

– `bool est_chiffre(int c)` qui vérifie si le caractère  $c$  est un chiffre (0 à 9). Cette fonction retourne `true` ou `false` suivant le cas.

Cette fonction existe dans la bibliothèque standard avec le prototype `int isdigit(int c)`. Elle est déclarée dans le fichier d'en-tête `ctype.h`. Consulter le manuel en ligne pour cette fonction.

– `bool est_lettre(int c)` qui vérifie si le caractère  $c$  est une lettre (minuscule ou majuscule). Cette fonction retourne `true` ou `false` suivant le cas.

**Exercice 5.** Selon le calendrier grégorien (à partir de l'année 1582), une année bissextile est une année dont le millésime est divisible par 4 à l'exception des années séculaires non multiples de 400 (une année séculaire est une année multiple de 100).

1. Écrire une fonction de prototype `bool est_gregorienne(int annee)` qui vérifie si une année est grégorienne (postérieure à 1581).
2. Écrire une fonction de prototype `bool est_seculaire(int annee)` qui vérifie si une année est séculaire.
3. Donner l'expression booléenne caractérisant l'année bissextile, puis écrire une fonction de prototype `bool est_bissextile(int annee)` qui vérifie si une année est bissextile.

## 2 Travaux Pratiques

Télécharger et décompresser l'archive contenant les fichiers tp.

**Exercice 6.** Compiler et exécuter (après avoir éventuellement apporté les corrections nécessaires) les fichiers correspondants aux exercices de la partie td. Vérifier que vous obtenez bien les résultats attendus.

**Exercice 7.** Écrire un programme `bissextile` qui à partir d'une année saisie à l'aide de la fonction `lire_entier()`, indique si cette année est bissextile ou non (pour le codage des expressions booléennes on utilisera le type booléen approprié défini dans `<stdbool.h>`).

ascii - Le jeu de caractères ASCII en octal, décimal, et hexadécimal.

DESCRIPTION

ASCII est l'acronyme de American Standard Code for Information Interchange. Il s'agit d'un code sur 7 bits. De nombreux codes sur 8 bits (tels que l'ISO 8859-1 le jeu de caractères par défaut de Linux) contiennent l'ASCII dans leur première moitié. L'équivalent international de l'ASCII est connu sous le nom de ISO 646.

La table suivante contient les 128 caractères ASCII.

Les séquences d'échappement '\X' pour les programmes C sont mentionnées.

Oct	Dec	Hex	Car	Oct	Dec	Hex	Car
000	0	00	NUL '\0'	100	64	40	@
001	1	01	SOH	101	65	41	A
002	2	02	STX	102	66	42	B
003	3	03	ETX	103	67	43	C
004	4	04	EDT	104	68	44	D
005	5	05	ENQ	105	69	45	E
006	6	06	ACK	106	70	46	F
007	7	07	BEL '\a'	107	71	47	G
010	8	08	BS '\b'	110	72	48	H
011	9	09	HT '\t'	111	73	49	I
012	10	0A	LF '\n'	112	74	4A	J
013	11	0B	VT '\v'	113	75	4B	K
014	12	0C	FF '\f'	114	76	4C	L
015	13	0D	CR '\r'	115	77	4D	M
016	14	0E	SO	116	78	4E	N
017	15	0F	SI	117	79	4F	O
020	16	10	DLE	120	80	50	P
021	17	11	DC1	121	81	51	Q
022	18	12	DC2	122	82	52	R
023	19	13	DC3	123	83	53	S
024	20	14	DC4	124	84	54	T
025	21	15	NAK	125	85	55	U
026	22	16	SYN	126	86	56	V
027	23	17	ETB	127	87	57	W
030	24	18	CAN	130	88	58	X
031	25	19	EM	131	89	59	Y
032	26	1A	SUB	132	90	5A	Z
033	27	1B	ESC	133	91	5B	[
034	28	1C	FS	134	92	5C	\ '\'
035	29	1D	GS	135	93	5D	]
036	30	1E	RS	136	94	5E	^
037	31	1F	US	137	95	5F	_
040	32	20	SPACE	140	96	60	'
041	33	21	!	141	97	61	a
042	34	22	"	142	98	62	b
043	35	23	#	143	99	63	c
044	36	24	\$	144	100	64	d
045	37	25	%	145	101	65	e
046	38	26	&	146	102	66	f
047	39	27	'	147	103	67	g
050	40	28	(	150	104	68	h
051	41	29	)	151	105	69	i
052	42	2A	*	152	106	6A	j
053	43	2B	+	153	107	6B	k
054	44	2C	,	154	108	6C	l
055	45	2D	-	155	109	6D	m
056	46	2E	.	156	110	6E	n
057	47	2F	/	157	111	6F	o
060	48	30	0	160	112	70	p
061	49	31	1	161	113	71	q
062	50	32	2	162	114	72	r
063	51	33	3	163	115	73	s
064	52	34	4	164	116	74	t
065	53	35	5	165	117	75	u
066	54	36	6	166	118	76	v
067	55	37	7	167	119	77	w
070	56	38	8	170	120	78	x
071	57	39	9	171	121	79	y
072	58	3A	:	172	122	7A	z
073	59	3B	;	173	123	7B	{
074	60	3C	<	174	124	7C	
075	61	3D	=	175	125	7D	}
076	62	3E	>	176	126	7E	~
077	63	3F	?	177	127	7F	DEL

HISTORIQUE

Une page de manuel ascii est apparue dans AT&T UNIX VERSION 7.

Sur les terminaux anciens, le code de soulignement (underscore) est affiché sous forme de flèche vers la gauche (backarrow), l'accent circonflexe (caret) est affiché sous forme de flèche vers le haut, et la barre verticale est interrompue en son centre.

Les caractères majuscules et minuscules ne diffèrent que par un bit, et le caractère Ascii "2" ne diffère du guillemet que d'un bit (ils partagent la même touche sur un clavier QWERTY) ceci facilite l'encodage mécanique ou par un clavier sans micro-contrôleur, comme sur les anciens télétypes.

Le standard ASCII a été publié par l'USASI (United States of America Standards Institute) en 1968.