

Priorité	Opérateur	Description	Exemple
0	()	appel de fonction, associativité	foo(); a = (b + c) * d;
	[]	indexation	int tab[3]; tab[0] = tab[1] = tab[2] = 0;
	.	nommage d'un champ	obj.cdr = NULL;
	->	nommage indirect de champ	pt->cdr = NULL;
1	!	négation	(!a) est vraie si a est fausse
	~	complément à 1	a & (~a) = 0
	-	opposé	
	++	incrémementation	i++; ++i;
	--	décrémementation	i--; --i;
	&	adresse	int i, *pt; pt = &i;
	*	valeur indirecte	*pt = 0; /* donne i = 0 */
	(type_de_donnée)	force le type (cast)	int i = (int)1.5;
	sizeof	taille en octets	i = sizeof i;
2	*	Multiplication	
	/	Division	
	%	Modulo	
3	+	Addition	
	-	Soustraction	
4	<<	Décalage à gauche	
	>>	Décalage à droite	
5	<	Strictement inférieur	
	<=	Inférieur ou égal	
	>	Strictement supérieur	
	>=	Supérieur ou égal	
6	==	Egal	
	!=	Différent	
7	&	“et” binaire	
8	^	“ou” exclusif binaire	
9		“ou” binaire	
10	&&	“et” logique	
11		“ou” logique	
12	? :	conditionnelle	c = (a < b) ? a : b;
13	= *= /= %= += -= ^= &= <<= >>= --=	Affectations	
14	,	Séquence	

TAB. 1 – Table des priorités des opérateurs C.

```

1  #include <stdio.h>
2  int main(void) {
3      char c;
4      c = getchar();
5      if (c >= 'a' && c <= 'z')
6          printf("Majuscule de %c : %c\n",c , c - 'a' + 'A');
7      else if (c >= 'A' && c <= 'Z')
8          printf("Minuscule de %c : %c\n",c , c + 'a' - 'A');
9      else
10         printf("Inconnu !\n");
11     return 0;
12 }

```

```

1  #include <stdio.h>
2  int main(void) {
3      char c;
4      int etat = 1;
5      while(1) {
6          if (etat == 1) {
7              printf("\nEntrer des lettres quelconques -> ");
8              etat = 0;
9          }
10         c = getchar();
11         if (c == '\n') {
12             if(etat == 0) {
13                 printf("Good bye !\n");
14                 break;
15             }
16             etat = 1;
17             continue;
18         } else
19             etat = 2;
20         if (c >= 'a' && c <= 'z')
21             printf("Majuscule de %c : %c\n",c , c - 'a' + 'A');
22         else if (c >= 'A' && c <= 'Z')
23             printf("Minuscule de %c : %c\n",c , c + 'a' - 'A');
24         else
25             printf("Caractere non-alphabetique !\n");
26     }
27     return 0;
28 }

```

FIG. 1 – Conversion Majuscule-Minuscule.

Lire le man des fonctions vues en cours. Exemple :

```

$ man 3 printf
$ man scanf
...

```

Pour obtenir la documentation du man, faire :

```

$ man man

```

Devoir 03 :

a. Écrire un programme qui affiche les caractères imprimables de la table ASCII à la manière de la commande `man ascii`.

b. Écrire, en utilisant la fonction `scanf` ou `fgets`, un programme qui transforme les minuscules d’une phrase en majuscules et vice-versa. Par exemple :
 (Bonjour, il est 8h00) → (bONJOUR, IL EST 8H00)

Le programme demande à l'utilisateur d’entrer une phrase puis la convertit et l’affiche. Il recommence tant que l'utilisateur entre une chaîne non vide. Dans l'exemple, vous remarquerez que les caractères non-alphabétiques n’ont pas été modifiés.

Faire attention aux entrées vides (ctrl-d).