

fonction	prototype	exemple	include	utilisation
<b>atoi</b>	int atoi( const char *chaine );	int i = atoi("2007");	stdlib.h	convertit une chaîne de caractères en entier
<b>srand</b>	void srand( unsigned seed );	srand(time(NULL));	stdlib.h	initialise le générateur de nombres aléatoires
<b>rand</b>	int rand( void );	int i = rand();	stdlib.h	retourne un nombre entier aléatoire entre 0 et <b>RAND.MAX</b>
<b>malloc</b>	void *malloc( size_t size );	int *tabInt = malloc(n*sizeof(int));	stdlib.h	alloue dynamiquement de la mémoire pour <b>size</b> octets
<b>free</b>	void free( void *ptr );	free(tabInt);	stdlib.h	libère de la mémoire allouée dynamiquement
<b>printf</b>	int printf( const char *format, ... );	printf("%d %d", x, y);	stdio.h	affiche à l'écran
<b>scanf</b>	int scanf( const char *format, ... )	scanf("%d %d", &x, &y);	stdio.h	saisit au clavier
<b>fopen</b>	FILE *fopen( const char *nomFichier, const char *mode );	FILE *f = fopen("fich.txt", "r");	stdio.h	ouvre le fichier nommé <b>nomFichier</b> dans le mode désiré (lecture, écriture, ...). Retourne un pointeur sur le fichier
<b>fclose</b>	int fclose( FILE *f );	fclose(f);	stdio.h	ferme le fichier <b>f</b>
<b>fprintf</b>	int fprintf( FILE *f, const char *format, ... );	fprintf(f, "%d %d", x, y);	stdio.h	écrit des informations formatées dans <b>f</b> retourne < 0 en cas d'erreur
<b>fscanf</b>	int fscanf( FILE *f, const char *format, ... )	int nb = fscanf(f, "%d %d", &x, &y);	stdio.h	lit dans <b>f</b> des informations formatées. Retourne le nombre d'objets convertis ou EOF
<b>fseek</b>	int fseek( FILE *f, long offset, int origin );	fseek(f, 0, SEEK.SET); fseek(f, 1024, SEEK.CUR);	stdio.h	positionne la tête de L/E dans <b>f</b>
<b>ftell</b>	long ftell( FILE *f );	long pos = ftell(f);	stdio.h	retourne la position de la tête de L/E de <b>f</b>
<b>fread</b>	int fread( void *buffer, size_t size, size_t num, FILE *f );	int nb = fread(tab, sizeof(int), n, f );	stdio.h	lit <b>num</b> objets de taille <b>size</b> dans <b>f</b> et les place dans <b>buffer</b> . Retourne le nombre d'objets lus
<b>fwrite</b>	int fwrite( const void *buffer, size_t size, size_t num, FILE *f );	int nb = fwrite(tab, sizeof(int), n, f );	stdio.h	écrit dans <b>f</b> <b>num</b> objets de taille <b>size</b> provenant de <b>buffer</b> . Retourne le nombre d'objets écrits
<b>fgets</b>	char *fgets( char *str, int num, FILE *f );	fgets(chaine, n, f);	stdio.h	lit au plus <b>num-1</b> caractères d'une ligne de <b>f</b> et les met dans <b>str</b>
<b>fflush</b>	int fflush( FILE *f );	fflush(stdout);	stdio.h	force l'écriture physique du buffer de sortie de <b>f</b>
<b>getchar</b>	int getchar( void );	int c = getchar();	stdio.h	retourne un caractère saisi au clavier
<b>putchar</b>	int putchar( int c );	putchar('a');	stdio.h	écrit un caractère à l'écran
<b>fgetc</b>	int fgetc( FILE *f );	int c = fgetc(f);	stdio.h	lit un caractère de <b>f</b>
<b>fputc</b>	int fputc( int ch, FILE *f );	fputc(car, f);	stdio.h	écrit le caractère <b>ch</b> dans <b>f</b>
<b>strncat</b>	char *strncat( char *destination, const char *source, size_t n );	strncat(chDest, chSource, n);	string.h	concatène au plus <b>n</b> caractères de <b>chSource</b> à la suite de <b>chDest</b> . <b>chDest</b> doit être assez grande
<b>strlen</b>	size_t strlen( char *str );	int lg = strlen(maChaine);	string.h	retourne la taille d'une chaîne
<b>strncpy</b>	char *strncpy( char *destination, const char *source, size_t n );	strncpy(chDest, chSource, n);	string.h	copie au plus <b>n</b> caractères d'une chaîne <b>source</b> dans une chaîne <b>destination</b>
<b>strchr</b>	char *strchr( const char *str, int c );	char *trouve = strchr(chaine, car);	string.h	retourne l'adresse de la première occurrence de <b>c</b> dans <b>str</b> ou NULL sinon
<b>clock</b>	clock_t clock(void);	clock_t start = clock();	time.h	retourne le temps CPU au moment de l'appel (le CPU compte <b>CLOCKS_PER_SEC</b> ticks par seconde)
<b>LireUneLigne</b>	int LireUneLigne( char *buf, int size, FILE* f );	int res = LireUneLigne(chaine, n, f);	-	version améliorée de <b>fgets</b> . Lit une ligne de <b>n</b> caractères maximum dans <b>f</b> et la range dans <b>chaine</b> . Retourne <b>FIN</b> si plus rien à lire, <b>OK</b> sinon