

Période 4 Les caractères et les chaînes de caractères

1/ Quelques fonctions sur les caractères ☼

Écrire les fonctions suivantes :

- **isAlpha** qui retourne 1 si l'argument *c* est une lettre (maj ou min) et 0 sinon.
int isAlpha (char c) ;
- **isAVow** qui retourne 1 si l'argument *c* est une voyelle (maj ou min) et 0 sinon.
int isAVow (char c)
- **isACons** qui retourne 1 si l'argument *c* est une consonne (maj ou min) et 0 sinon.
int isACons (char c) ;
- **isADigit** qui retourne 1 si *c* est un chiffre et 0 sinon.
int isADigit (char c) ;
- **toLower** qui retourne le code ASCII de *c* en minuscule si *c*'est une lettre, -1 sinon.
int toLower (char c) ;
- **toUpper** qui retourne le code ASCII de *c* en majuscule si *c*'est une lettre, -1 sinon.
int toUpper (char c) ;
- **letAleat** qui retourne une lettre minuscule aléatoire comprise entre a et z.
char letAleat (void) ;

2/ Fonctions sur les chaînes de caractères ☼

Écrire les fonctions suivantes :

stringUpper qui modifie la casse de la chaîne de caractère passée en argument en majuscule. Les lettres majuscules et les symboles ne sont pas modifiés. Cette fonction retourne le nombre de caractères modifiés : **int stringUpper(char *s) ;**

voyelles qui retourne le nombre de voyelles présentes dans la chaîne *s*

int voyelles (const char *s) ;

lettres qui retourne le nombre de lettres, quelque soit la casse, présentes dans la chaîne *s*.

int lettres (const char *s) ;

zero qui remplit avec des valeurs égales à '\0' les *n* premiers octets de la chaîne *t*

void zero (char *t, int n) ;

compt qui compte et retourne le nombre de fois où le caractère *orig* est présent dans la chaîne *s*.

int compt (const char *s, char orig) ;

replace qui remplace le caractère *src* par le caractère *dst* dans la chaîne *s*. Cette fonction retourne le nombre de remplacements effectués

int replace (char *s, char src, char dst) ;

truncate qui supprime les n derniers caractères de la chaîne s. Cette fonction retourne le nombre de caractères effectivement supprimés. ☼☼

```
int truncate (char *s, int n) ;
```

copyNoChar qui recopie la chaîne src dans dst à l'exception du caractère c. Cette fonction retourne le nombre de caractères effectivement copiés. ☼☼

```
int copyNoChar (const char *src, char *dst, char c) ;
```

reverse qui recopie la chaîne src dans dst à l'envers. ☼☼☼

```
void reverse (const char *src, char *dst) ;
```

2/ Génération de mot de passe aléatoires ☼☼

Écrire les fonctions suivantes :

password1 qui génère le mot de passe pass qui contient n lettres minuscules aléatoires.

```
void password1 (char * pass, int n) ;
```

password2 qui génère le mot de passe pass qui contient n lettres, chaque lettre aléatoire ayant une équiprobabilité d'être minuscule ou majuscule.

```
void password2 (char * pass, int n) ;
```

password3 qui génère le mot de passe pass qui contient n caractères imprimables aléatoires.

```
void password3 (char * pass, int n) ;
```

NB : utiliser la fonction `isprint` de `ctype.h`

3/ Conversions chaîne ↔ nombre ☼☼

Écrire les fonctions suivantes :

toInt qui retourne la valeur entière représentée par la chaîne s composée de caractères nombres décimaux [0;9] et éventuellement d'une indication de signe '+' ou '-':

```
int toInt( const char * s ) ;
```

hexToInt qui retourne la valeur entière représentée par la chaîne s composée de caractères hexadécimaux [0;9] et [A;F] :

```
int hexToInt( const char * s ) ;
```

binToInt qui retourne la valeur entière représentée par la chaîne s composée de caractères 0 ou 1 :

```
int binToInt( const char * s ) ;
```

toFloat qui retourne la valeur décimale représentée par la chaîne s de caractères nombres décimaux [0;9] et éventuellement d'un point décimal ',' ou '.' et/ou d'une indication de signe '+' ou '-' :

```
float toFloat( const char * s ) ; ☼☼☼
```