Rendu de libft : Comment j'ai fait mes fonctions

Partie une : Fonctions de la libc

A) Fonctions ne nécessitant pas de fonctions externes:

Isalpha

Je me suis servie de la table ascii pour me permettre de vérifier que le caractère en paramètre possède une valeur entre celle de la lettre a et la lettre z (en majuscule ou non) en plaçant le tout dans une condition. Si la condition est vraie, elle retourne un, sinon elle retourne 0.

Isdigit

J'ai fait la même chose que pour isalpha mais avec les chiffres.

Isalnum

Il s'agit des fonctions isalpha et isdigit combinées.

isascii

J'ai vérifié que la valeur décimale du caractère pris en entrée soit comprise entre 0 et 127, qui sont les limites de valeur décimales de la table ascii.

isprint

J'ai vérifié que le caractère en entrée est printable

Toupper

J'ai vérifié que le caractère en entrée est bien une lettre minuscule et je me suis servie de la table ascii pour ajouter 32 a la valeur décimale pour que celui-ci devienne la même lettre mais en majuscule, sinon la fonction renvoie le char.

Tolower

Pareil que toupper mais en enlevant à la valeur décimale 32 pour que la lettre devienne une minuscule

Strchr

J'ai commencé par déclarer une variable de type size_t et l'instancier à 0. Celle-ci va me permettre de me ballader dans la chaine de caractères passée en paramètre à l'aide d'une boucle while qui a pour condition d'arrêt tant que la position dans le tableau (à l'indice i) soit différente du caractère c passé en paramètre. si j'atteins la fin de la chaine (verifié par le fait que le tableau a l'indice i ne soit pas egal a \0) . Si je sors de la premiere boucle je renvoie un pointeur vers ce caractère casté en char*.

Strrchr

Meme principe qu'avec strchr mais en donnant à la variable j la valeur de la taille de la chaine grace a la fonction strlen, tant que j >= 0 dans la boucle while et en enlevant -1 a j pour chaque passage de boucle. si j atteint 0 on retourne null.

strncmp

Strlen
J'ai commencé par créer une variable de type size_t et initialisée a 0 me permettant de me ballader dans le tableau, puis j'ai fait une variable selon laquelle on ne sortirait pas du tableau et j'ai ajouté un à i a chaque fois que l'on passait dans celle ci. Puis j'ai retourné cette variable.

Memset
Comme cette fonction a pour paramètre un pointeur générique, j'ai commencé par créer une variable de type tableau de char et je lui ai donné la valeur de ce pointeur. j'ai également créé une variable de type size_t et je l'aie initialisée à 0. j'ai ensuite créé une boucle while avec pour condition i < à la taille de la mémoire passée en paramètre et j'ai donné à la position dans le tableau pour valeur le caractère en paramètre.

Memchr
Même chose que pour strchr mais en castant le pointeur générique en unsigned char * et en conditionnant la sortie de la boucle à i < n car il n'y a pas de \0.

Memcmp

Bzero
J'ai utilisé la fonction memset pour remplir la mémoire en paramètre de \0

Strnstr
J'ai commencé par créer deux variables de type size_t initialisées à 0. Ensuite, je me suis assurée que little 'etait pas vide, si c'etait le cas je retournais simplement big, puis je m'assurais aussi que big n'etait pas plus petit que little, si c'etait le cas je retournais null, et sinon je crée une boucle dans laquelle je ne sors pas de big avec i et que i est inferieur a len.

ensuite je fais une seconde boucle while dans laquelle je ne sors pas de lit avec j et que ensuite je me ballade dans big pour verifier a l'aide de j que i+j sont bien egaux a lit a l'indice j. je verifie que je ne depasse pas len en additionnant i + j. et j'ajoute un a j a chaque fin de boucle.

Si grace a cette boucle, on atteint la fin de la boucle avec j, je retourne l'indice de la premiere lettre de lit dans big .

Si je sors des deux boucles sans avoir trouvé lit, je renvoie null.

Memcpy
J'ai commencé par créer une variable de type size_t me permettant de me ballader dans le tableau et deux pointeurs vers deux tableaux de char dont un de type constant . Ensuite j'ai vérifié que dest et src ne sont pas egaux a null car dans ce

cas je dois renvoyer null, et si ce n'est pas le cas , je créé une boucle selon laquelle i ne depasse pas la taille de src et je remplis la memoire de dest avec la valeur de src. grace a i.

Atoi
Je me sers de la première fonction pour passer tous les caractères vides de la chaine, et j'attribue la valeur du int ainsi envoyé a un int qui se balladera dans ma chane . ensuite je verifie que mq chqine ne contient pqs de - et si c'est le cas je donne a nb une valeur positive, puis j'utilise ft transform pour convertir ma chaine en int a l'aide de la table ascii.
Memmove
Strlcpy
Strlcat

B) Fonctions utilisant malloc

Calloc
strdup

Partie deux : Fonctions supplémentaires:

Substr
Strjoin
Strtrim
Split
Itoa
Strmapi
Striteri
Putchar_fd
Putstr_fd
Putendl_fd
putnbr_fd

Partie Bonus :

Lstnew
Lstadd_front
Lstsize
Lstlast
Lstaddback