Responsable pédagogique	AF	AM	OP	**PM**
Période	**Sem1**	Sem2	Sem3	Sem4
Volume horaire	Cours/TD		TP	
			6	

ESSTD
TP getchar,putchar,scanf,printf

Indicateur temporel (hors rédaction du compte-rendu) :

questions	1h	2h	3h	4h	5h	6h
1						
2						
3						

Tous les fichiers de ce TP se trouvent dans le répertoire /pub/ESSTD de la machine OUVEA (dans votre voisinage réseau)

Cahier des charges

Le but est de coder plusieurs programmes effectuant des E/S standard type getchar(), putchar() ; mais aussi printf(), scanf().

1. Création d'un getchar personnel

Vous allez coder dans un source nommé mygetchar.c, la lecture sur le canal d'entrée standard (clavier).
Le rôle de ce programme est de saisir un caractère au clavier, sauf le CTRL-D (séquence d'échappement type fin de fichier) .

Déclarer une variable lettre de type char ; puis utilisez l'appel read() en codant *read(0, &lettre , 1) ;*
Codez de suite un affichage ensuite de lettre avec *write(1,&lettre,1) ;*

Testez, compilez en faisant gcc -Wall -o mygetchar mygetchar.c
Lancer votre exécutable ./mygetchar

Qu'observez-vous dans le terminal ?
.....
.....
.....

Modifiez légèrement votre source mygetchar.c, pour pouvoir sortir du programme dès que le EOF (End Of File) est généré par CTRl-d .
Indice : read() retourne 0 si EOF est détecté .

Ligne de code ..... if(read(0,&lettre,1) == 0){return 1;}

Remplacez maintenant, votre ligne de code read(0, &lettre,1) ; par l'appel à la fonction scanf() ;
où vous mettrez le bon format%c en argument.
Ligne de code : scanf("%c",&lettre);

Testez, compilez en faisant gcc -Wall -o mygetchar2 mygetchar2.c
Lancer votre exécutable ./mygetchar2

Qu'observez-vous dans le terminal ?

On observe le même résultat que précedemment lorsque l'on appui sur ctrl + D cela

Modifiez le%c en%d dans le scanf.
Retestez.Qu'observez-vous dans le terminal ?

On observe qu'il affiche le chiffre qu'une seule fois

Qu'elle différence majeure y-a t il, d'après vos 2 programmes, entre la fonction read(), et la fonction scanf() ? Qu'offre le scanf, p/r a getchar() ?

Le scanf permet d'afficher des chiffres et des lettres et le scanf permet permet de prendre plusieurs char contrairement au getchar qui prend char par char

2. Création d'un putchar personnel

Créer un source nommé cette fois myputchar.c, à partir du source mygetchar.c.
Faites une copie. En exécutant la commande cp mygetchar.c putchar.c
Vous remplacerez maintenant votre ligne de code write(1, &lettre,1) ; par un putchar();

Qu'observez-vous dans le terminal ?
le char s'affiche de la même manière qu'avec le mygetchar2.c

Codez un 2eme source nommé cette fois myputchar2.c, où vous afficherez en décimal et ascii le caractère saisi. Vous allez utilisez l'appel printf(), en passant un formatage en decimal et caractère à la fois.
Pour avoir par exemple l'exécution ci-dessous

```
R
lettre en ascii R  en decimal 82
```

Ligne de code ..printf("Lettre en ascii %c en decimal %d",lettre,lettre);

Testez, compilez en faisant gcc -Wall -o myputchar2 myputchar2.c
Lancer votre exécutable ./myputchar2

Qu'elle différence majeure y-a t il, d'après vos 3 programmes, entre la fonction putchar(), et printf() ?

la différence entre putchar et printf est qu'avec printf on peu afficher du décimal et des char contrairement au putchar qui affiche soit l'un soit l'autre

3. Les chaines de caractères

Vous le savez, elles sont toutes terminées par le caractère NUL (0 en code ascii).

Pour les saisir, et les afficher, nous allons créer des fonctions myscanf, myprintf

myscanf avec getchar()

Codez en partant de mygetchar.c (celui avec le read() !! et pas le scanf()...) :
Créer bien sûr un nouveau source par exemple : *myscanf.c*
On demande une boucle de saisie, de plusieurs caractères y compris avec des espaces.

Code (Algorithme proposé)

```
TANT que lettre <> CRLF  FAIRE
        buffer [ i ] ← lettre ;
        i ← i + 1 ;
        saisir(lettre) ;
FINTQ
        buffer [ i ] ← 0 ;
```

.....
.....
.....
.....
.....
.....
.....
.....

Faites vos essais.
Pensez à affichez la chaine avec un printf("%s " , buffer) ;

Faites un 2eme source myscanf2.c. Cette fois remplacez toute votre boucle par scanf("%s ",buffer) ;
Compiler et tester avec des mots avec et sans espaces.

Qu'observez-vous ?

On peux voir que la condition d'arrêt de la boucle n'est plus respecter on est donc dans une boucle infini

.....
.....
.....
.....
.....

myprintf

Codez en partant de myputchar.c (celui avec le write() !! ...) :
Créer bien sûr un nouveau source par exemple : *myprintf.c*
On demande une boucle d'affichage, de plusieurs caractères y compris avec des espaces.

Code (Algorithme proposé)

```
TANT que lettre <> 0  FAIRE
        Afficher lettre ;
        i ← i + 1 ;

FINTQ
```

.....
.....
.....
.....
.....

Compiler et tester avec des mots avec et sans espaces.

Qu'observez-vous ?

On peux voir qu'il affiche correctement la chaine de char qu'on lui impose

Synthèse (oui / non)

fonction	Saisie possible en caractère seul	Saisie possible en différents formats (décimal ...)	Saisie de chaines avec espaces	Saisie de chaines sans espaces	Affichage sous forme caractère seul	Affichage sous plusieurs formats possibles	Affichage de chaines avec espaces	Affichage de chaines sans espaces
getchar								
putchar								
scanf								
printf								