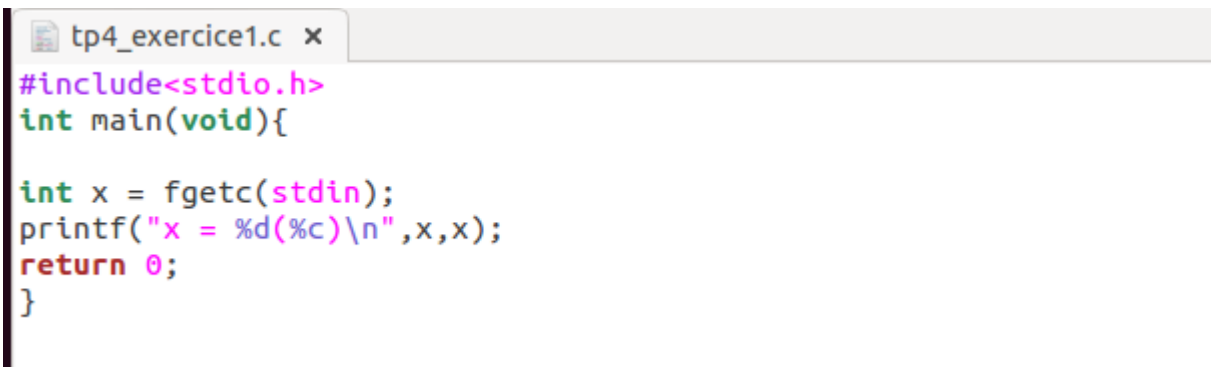


TRAVAUX PRATIQUE N° 4 (ENTREE/SORTIE)

Travail 1: Après exécution du programme ci-dessous que remarquez-vous lorsque après la demande de l'entrée standard vous faites :

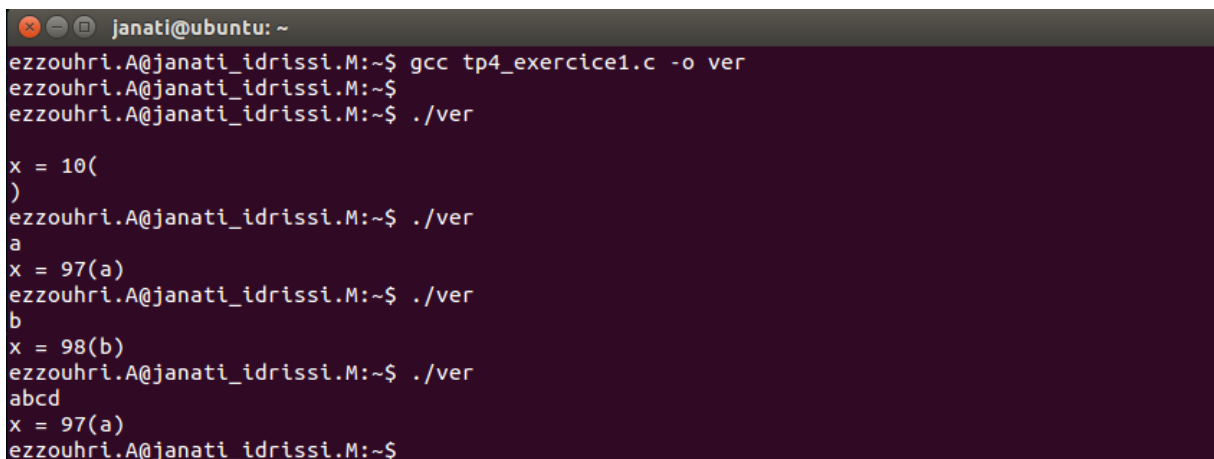
<enter>
a<enter>
a<backspace>b<enter>
abcd<enter>

```
#include <stdio.h>
int main (void)
{
    int x = fgetc(stdin);
    printf ("x = %d ('%c')\n", x, x);
    return 0;
}
```



```
tp4_exercice1.c x
#include<stdio.h>
int main(void){

int x = fgetc(stdin);
printf("x = %d('%c')\n",x,x);
return 0;
}
```



```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gcc tp4_exercice1.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver
x = 10(
)
ezzouhri.A@janati_idrissi.M:~$ ./ver
a
x = 97(a)
ezzouhri.A@janati_idrissi.M:~$ ./ver
b
x = 98(b)
ezzouhri.A@janati_idrissi.M:~$ ./ver
abcd
x = 97(a)
ezzouhri.A@janati_idrissi.M:~$
```

Observation :

<enter>

Lorsqu'on clique sur la touche « entrer », on remarque que le programme affiche le code ASCII de \n qui égale à 10 , et entre les parenthèses le \n prend sa valeur et fait une retour à la ligne.

a<enter>

la même chose se déroule pour l'affichage de caractère saisi et son code ASCII.

a<backspace>b<enter>

ici le caractère extrait est b.

abcd<enter>

lorsqu'on saisie une chaîne de caractère, le programme prend seulement le premier et l'affiche sur la sortie standard.

Travail 2: *En utilisant la fonction `getc()` ou `getchar()`, écrire un programme C qui permet de lire un caractère à partir de l'entrée standard et de l'afficher sur la sortie standard.*

```
*tp4_exercice2.c x
#include<stdio.h>
#include<stdlib.h>
int main(){
    printf("entrez un caractère :");
    char c = getc(stdin);
    printf("le caractère saisi est:");
    putc(c,stdout);
    printf("\n");
}
```

```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit tp4_exercice2.c
ezzouhri.A@janati_idrissi.M:~$ gcc tp4_exercice2.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver
entrez un caractère :s
le caractère saisi est:s
ezzouhri.A@janati_idrissi.M:~$
```

```
tp4_exercice2-2.c x
#include<stdio.h>
#include<stdlib.h>
int main(){
    char c;
    printf("entrez un caractère :");
    c=getchar();
    printf("le caractère saisi est : ");
    putchar(c);
    printf("\n");
}
```

```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit tp4_exercice2-2.c
ezzouhri.A@janati_idrissi.M:~$ gcc tp4_exercice2-2.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver
entrez un caractère :g
le caractère saisi est : g
ezzouhri.A@janati_idrissi.M:~$
```

Travail 3: En utilisant la fonction `scanf()`, écrire un programme C qui permet de lire une chaîne de caractères ainsi que le nombre de ces caractères à partir de l'entrée standard et de les afficher sur la sortie standard.

```
tp4_exercice3.c x
#include<stdio.h>
#include<stdlib.h>

int main(){

    char ch[100];
    int n;
    printf( "saisie la chaine de caractère + leurs nombres :");
    scanf("%s %d",ch,&n);

    printf( "\n la chaine saisie et le nombre : %s %d ",ch,n);
    return 0;

}
```

```
janati@ubuntu: ~
ezzouhri.A@janati_idrissi.M:~$ gedit tp4_exercice3.c
ezzouhri.A@janati_idrissi.M:~$ gcc tp4_exercice3.c -o ver
ezzouhri.A@janati_idrissi.M:~$ ./ver
saisie la chaine de caractère + leurs nombres :jesuisétudiante 16

la chaine saisie et le nombre : jesuisétudiante 16
ezzouhri.A@janati_idrissi.M:~$
```

Travail 4: Ecrire un programme C correspondant à la commande Shell :
`Cat fichier1>>fichier2.`

```
tp4_exercice4.c x a.c x b.c x
#include<stdio.h>
#include<stdlib.h>
main(int n,char *v[])
{
    FILE *f1 , *f2;
    char c;
    if(n != 3)
    {
        printf("nombre de parametre incorrect \n");
        exit(1);
    }
    if ((f1 = fopen(v[1],"r")) == NULL){
        printf("ouverture impossible de %s \n",v[1]);
        exit(2);}
    if ((f2 = fopen(v[2],"a")) == NULL){
        printf("ouverture impossible de %s\n",v[2]);
        exit(3);}
    while((c=getc(f1)) != EOF ) putc(c,f2);
    fclose(f1);
    fclose(f2);
}
```

```
janati@ubuntu: ~  
ezzouhri.A@janati_idrissi.M:~$ gedit a.c  
ezzouhri.A@janati_idrissi.M:~$ gedit b.c  
ezzouhri.A@janati_idrissi.M:~$ cat a.c  
hello world!!!  
ezzouhri.A@janati_idrissi.M:~$ cat b.c  
we are students .  
ezzouhri.A@janati_idrissi.M:~$ gedit tp4_exercice4.c  
ezzouhri.A@janati_idrissi.M:~$ gcc tp4_exercice4.c -o ver  
ezzouhri.A@janati_idrissi.M:~$  
ezzouhri.A@janati_idrissi.M:~$ ./ver  
nombre de parametre incorrect  
ezzouhri.A@janati_idrissi.M:~$ ./ver b.c a.c  
ezzouhri.A@janati_idrissi.M:~$ cat b.c  
we are students .  
ezzouhri.A@janati_idrissi.M:~$ cat a.c  
hello world!!!  
we are students .  
ezzouhri.A@janati_idrissi.M:~$
```

Travail 5: *Ecrire un programme appelé `ECRI_TEXT` qui permet de placer du texte dans le fichier `test.txt` en utilisant les fonctions `fprintf()` et `fputs()`. Vérifier que l'ouverture de `test.txt` n'a pas connue d'erreur.*

Travail 6 : *(Suite de l'exercice 5). Affichez sur la sortie standard le contenu du fichier `test.txt` en utilisant les fonctions `fscanf()` et `fgets()`. Vérifier que l'ouverture de `test.txt` n'a pas connue d'erreur.*