

```
// EXEMPLE 1 :

export FT_NBR1=\\'\\?\\\\"\\'\\

// on initialise la variable FT_NBR1 avec \'?"'\

// on veut prendre le nombre contenu dans la variable FT_NBR1 qui est en base \'"?!

// FT_NBR1 est en base \'"?!
// cela signifie que :
// ' = 0
// \ = 1
// " = 2
// ? = 3
// ! = 4

echo $FT_NBR1 | tr "\'\\\\\\\\\\"?\\!" "01234"

// on remplace ' par 0, \ par 1, " par 2, ? par 3 et ! par 4
// ATTENTION : pour échapper \ lui-même, il faut utiliser 4 antislashes !
// en effet, le shell interprète les caractères d'échappement
// AVANT de passer la commande à tr
// le premier niveau d'échappement est pour le shell
// le deuxième niveau d'échappement est pour tr
// (car tr reçoit ce qui a été interprété par le shell)

// ainsi, pour que le shell passe \\ à tr
// (qui sera interprété comme un seul \ par tr)
// il faut écrire \\\\ dans la commande shell
// le shell réduira donc \\\\ à \\
// tr recevra donc \\ comme argument
// lui indiquant de chercher le caractère \

// RESULTAT : 10321201
// \'?"'\ sera traduit par
// 10321201
```

```
export FT_NBR2="rcrdmddd"
```

```
// on initialise la variable FT_NBR2 avec rcrdmddd
```

```
// on veut prendre le nombre contenu dans la variable FT_NBR2 qui est en base mrdoc
```

```
// FT_NBR2 est en base mrdoc
```

```
// cela signifie que :
```

```
// m = 0
```

```
// r = 1
```

```
// d = 2
```

```
// o = 3
```

```
// c = 4
```

```
echo $FT_NBR2 | tr "mrdoc" "01234"
```

```
// on remplace m par 0, r par 1, d par 2, o par 3 et c par 4
```

```
// RESULTAT : 14120222
```

```
// rcrdmddd sera traduit par
```

```
// 14120222
```

```
// on combine les commandes
```

```
// echo $FT_NBR1 | tr "'\\\\\\\\\\\\\\\\'\\?\\!" "01234"
```

```
// et
```

```
// echo $FT_NBR2 | tr "mrdoc" "01234"
```

```
// pour pouvoir calculer la somme de la valeur numérique de FT_NBR1 et de celle de FT_NBR2
```

```
// pour cela, peut utiliser des sous-shells pour afficher le résultat de chacune des 2 commandes
```

```
// puis afficher l'addition du résultat des 2 commandes
```

```
// (on cherche à obtenir le résultat : 10321201 + 14120222)
```

```
// qui sera ensuite acheminé vers bc pour effectuer le calcul (voir plus bas)
```

```
// pour exécuter une commande dans un sous-shell, on écrit $(commande)
```

```
// (en remplaçant "commande" par la commande correspondante)
```

```
// on écrit donc :
```

```
// $(echo $FT_NBR1 | tr "'\\\\\\\\\\\\\\\\'\\?\\!" "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234")
```

```
// en n'oubliant pas de faire un echo de ce résultat
```

```
// ainsi, la commande sera :
```

```

echo $(echo $FT_NBR1 | tr "'\\"\\\\\\""?\\!" "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234")

// RESULTAT : 10321201 + 14120222

// on transmettra ensuite ce résultat à la commande bc
// pour qu'elle effectue le calcul
// on veut ensuite afficher la somme de ces deux nombres en base gtaio luSnmef

// ATTENTION :
// FT_NBR1 est en base '\\"?!'
// donc en base 5 numériquement
// car 5 symboles caractères (' \ " ? et !) sont utilisés
// pour donner 5 symboles numériques (0 1 2 3 et 4)

// FT_NBR2 est en base mrdoc,
// donc en base 5 numériquement
// car 5 symboles caractères (m r d o et c) sont utilisés
// pour donner 5 symboles numériques (0 1 2 3 et 4)

// on veut obtenir un résultat en base gtaio luSnmef,
// donc en base 13 numériquement
// car 13 symboles caractères (g t a i o ' ' l u S n m e et f) sont utilisés
// pour donner 13 symboles (0 1 2 3 4 5 6 7 8 9 A B C)
// (ATTENTION : le symbole espace correspond au symbole 5)

// on utilise la commande bc pour faire la somme, en base 5,
// des valeurs numériques de FT_NBR1 et de FT_NBR2
// puis pour convertir cette somme en base 13

// pour cela, on indique tout d'abord à la commande bc
// les données utilisées pour le calcul
// on veut faire la somme des deux nombres
// contenus dans les variables FT_NBR1 et FT_NBR2
// donc :
// 10321201 + 14120222
// ce qui est le résultat que l'on a obtenu avec la commande précédente

```

```
// pour passer ce résultat (la sortie de la commande précédente)
// en argument de la commande suivante
// on utilise xargs
// (car la commande bc ne lit pas l'entrée standard)
// donc la seule utilisation d'un pipe (|) ne fonctionnerait pas ici
```

```
// la base dans laquelle on veut convertir le résultat du calcul
// avec obase (output base) :
// on veut convertir le résultat du calcul en base 13
// on écrit donc obase=13;
```

```
// puis la base dans laquelle sont les valeurs utilisées pour le calcul
// avec ibase (input base)
// les valeurs utilisées pour le calcul sont en en base 5
// on écrit donc ibase=5;
```

```
// ATTENTION : le fait d'indiquer obase avant ibase est important ici
// pour que la conversion de sortie utilise la nouvelle base (la base 13)
```

```
echo $(echo $FT_NBR1 | tr "'\\\\\\\\\"'?"'!' "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234") | xargs echo 'obase=13;
ibase=5;' | bc
```

```
// RESULTAT : 82671
// enfin, on convertit ce résultat en base gtaio luSnmef
// cela signifie que :
// g = 0
// t = 1
// a = 2
// i = 3
// o = 4
// ' ' = 5
// l = 6
// u = 7
// S = 8
// n = 9
// e = A
// m = B
// f = C
```

```

// on utilise donc la commande tr '0123456789ABC' 'gtaio luSnemf'
// pour convertir le résultat en base gtaio luSnemf
// ATTENTION : on effectue ici la traduction inverse
// de la base 13 vers la base gtaio luSnemf

echo $(echo $FT_NBR1 | tr "'\\\\\\\\\"'?"'!' "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234") | xargs echo 'obase=13;
ibase=5;' | bc | tr '0123456789ABC' 'gtaio luSnemf'

// RESULTAT : Salut
// 82671 sera traduit par
// Salut

// EXEMPLE 2 :

export FT_NBR1='\\\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\"'

echo $FT_NBR1 | tr "'\\\\\\\\\"'?"'!' "01234"

// RESULTAT : 1212412124121241212412124121241212
// \\\\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\"!\\\\" sera traduit par
// 1212412124121241212412124121241212

export FT_NBR2='dcrmcmmoododmrrrmorcmcrmomo'

echo $FT_NBR2 | tr "mrdoc" "01234"

// RESULTAT : 2414040333232011103140410303
// dcrmcmmoododmrrrmorcmcrmomo sera traduit par
// 2414040333232011103140410303

echo $(echo $FT_NBR1 | tr "'\\\\\\\\\"'?"'!' "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234")

// RESULTAT : 12124121241212412124121241212 + 2414040333232011103140410303

```

```
echo $(echo $FT_NBR1 | tr "'\\\\\\\\\\\\\\\\\\'\\?\\!" "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234") | xargs echo 'obase=13;ibase=5;' | bc
```

```
// RESULTAT : 8A0BA91213495C2761
```

```
echo $(echo $FT_NBR1 | tr "'\\\\\\\\\\\\\\\\\\'\\?\\!" "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234") | xargs echo 'obase=13;ibase=5;' | bc | tr '0123456789ABC' 'gtaio luSnemf'
```

```
// RESULTAT : Segmentation fault  
// 8A0BA91213495C2761 sera traduit par  
// Segmentation fault
```

```
vim add_chelou.sh
```

```
<i>
```

```
#!/bin/sh
```

```
echo $(echo $FT_NBR1 | tr "'\\\\\\\\\\\\\\\\\\'\\?\\!" "01234") + $(echo $FT_NBR2 | tr "mrdoc" "01234") | xargs echo 'obase=13;ibase=5;' | bc | tr '0123456789ABC' 'gtaio luSnemf'
```

```
<escape>
```

```
:wq
```

```
<enter>
```

```
chmod +x add_chelou.sh
```

```
./add_chelou.sh
```