

TP1b

On va s'intéresser au programme vu en TD.

Le but est le suivant :

- affiche le répertoire courant ;
- va dans le sous-répertoire `toto` ;
- affiche de nouveau le répertoire courant.

1 en shell

Question 1 *Indiquer les commandes successives à entrer dans le shell pour faire ces trois actions.*

Question 2 *Que se passe-t-il si le dossier `toto` n'existe pas ? Si `toto` est un fichier ?*

Écrire maintenant ces trois commandes sur une ligne en les séparant d'abord par des ; et ensuite par des `&&`. Regarder les résultats selon si le dossier `toto` existe ou pas.

Question 3 *Qu'observez vous ? En déduire le comportement de ~~gcc~~.*

2 En C

On se donne le fichier C suivant.

```
#include <stdio.h>
#include <unistd.h>
#define MAXP 200
int main(void) {
    char s[MAXP],t[MAXP];
    printf("Je commence");

    getcwd(s,MAXP);
    chdir("toto");
    getcwd(t,MAXP);

    printf("J'étais dans %s\n",s);
    printf("Je suis arrivé dans %s\n",t);
    printf("J'ai fini\n");
}
```

Question 4 *Compiler le (à l'aide de la commande `gcc`) et exécuter le.*

Question 5 *Que se passe-t-il si le dossier `toto` n'existe pas ? Comment corriger le bug ? (on ne demande pas de donner le code C mais juste d'indiquer les modifications à faire).*

3 En python

Question 6 *En se servant de ce qui a été vu en TD, écrire le script python.*

Question 7 *Le tester et vérifier son bon fonctionnement dans le cas où le dossier toto existe.*

Question 8 *Ajouter la gestion d'erreur dans le cas où l'appel système échoue.*

Question 9 *Améliorer cette gestion d'erreur pour distinguer entre les cas où le dossier n'existe pas et le cas où toto est un fichier.*