Recommandations : c'est du système, pensez bien aux tests des primitives. Évitez les mallocs, inutile de complexifier. En cas de besoin, servez vous du man.

Question 1 : écrire la commande `toUpper`.

La commande **`toUpper`** lance le programme qui est passé en argument et affiche son résultat en MAJUSCULES
Exemple:

```
$ who
sylvain
$ who | ./toUpper
SYLVAIN
$ date | ./toUpper
WED JUL 06 17:12:12 CEST 2016
```

Question 2 : Le plus grand d'entre nous !

Écrire un programme déterminant quel est le plus grand numéro de PID possible pour un processus (PS : créez vraiment les processus, et surveiller les résutats. Quand la limite est atteinte, le système recherche le premier numéro de PID disponible en repartant de 2)

Question 3 : I need protection

Le lancement de **gedit** à partir d'un **xterm** pose parfois le problème suivant : Certains utilisateurs font un CTRL Z sur le **shell** , ce qui met **gedit** en suspend (et rend la main sur le shell). Ecrire une version *protectedGedit*, le plus simplement possible, qui n'a pas ce problème, c'est à dire qui ignore le signal **SIGSTP**.

Question 4 le pouvoir de la source

Ecrire un programme prenant en argument un *fichier .c* et essayant de le compiler avec **gcc**. Si la compilation s'est bien passée, le programme doit alors exécuter le programme *./a.out* obtenu. Sinon, affichez un message d'erreur (« *Erreur de compilation* »).

Bonus : le **gcc** ne doit rien afficher.

Bonus 2 : et si on donne plusieurs fichiers sources (chacun indépendant) ?

Question 5: File Info

Ecrire un programme qui parcourt le répertoire dont le nom est donné en argument, et affiche, pour chaque nom de fichier, son numéro d'Inode et si il s'agit d'un fichier, d'un répertoire, ou d'autre chose.