Commander la LED avec un bouton poussoir

Cahier des charges : Lorsqu'on appuie sur le bouton poussoir, la LED s'allume et reste allumée une fois le bouton relâché. Le prochain appui sur le bouton poussoir éteindra la LED et ainsi de suite.
Ouvrez l'éditeur de texte `nano` :

```
pi@raspberrypi ~ $ nano cde_LED_02.sh
```

Saisissez le script destiné à lire l'état d'un bouton poussoir connecté sur le GPIO16 et à commander une LED connectée sur le GPIO 20.

```
#!/bin/bash
# Script cde_LED_02.sh
# Lire l état d un bouton poussoir
# Et commander une LED
# A chaque appui, l'état de la LED change (toggle)
# Ce script ne comporte pas de test destiné à l interrompre
# Il s arrête en appuyant sur CTRL  C

# Effacer l'écran
clear

# Rendre le répertoire /sys/class/gpio actif
cd /sys/class/gpio

################ Configuration port GPIO du bouton poussoir

# Créer l accès au port GPIO 16
# Pour lire l état du bouton poussoir
# Ne rien faire s il existe déja
if [ -d "gpio16" ]; then
 echo "gpio16 existe dejà"
else
 echo "gpio16 : Création"
 echo 16 > export
fi

# Rendre le répertoire gpio16 actif
cd gpio16/

# Configurer le port GPIO 16 en entrée
# Normalement il y est par défaut… mais on ne sait pas ce qui a pu se passer avant
echo in > direction

################ Configuration port GPIO de la LED

# revenir dans le dossier /sys/class/gpio
cd ..

# Créer l accès au port GPIO 20
# Pour commander la LED
# Ne rien faire s il existe déja
if [ -d "gpio20" ]; then
 echo "gpio20 existe dejà"
else
 echo "gpio20 : Création"
 echo 20 > export
```

```
fi

# Rendre le répertoire gpio20 actif
cd gpio20/

# Configurer le port GPIO 20 en sortie
echo out > direction

# Eteindre la LED
echo 0 > value

############################  Boucle de scrutation du port GPIO 16

# Initialiser la variable x à 1 pour la boucle principale
x="1"

# initialiser LED éteinte
led="0"

# Boucle while : cette boucle s exécute tant que
# la condition est vérifiée
# ici la condition est toujours vraie :
# la boucle ne s interrompt jamais
# Il faudra en sortir avec un CTRL C

while [ $x -gt 0 ]
# Le bloc inclus entre do et done est exécuté
# par la boucle while
# c est la boucle principale

# Début du bloc d instructions
do
    # Envoyer la valeur sur la LED
    echo $led > /sys/class/gpio/gpio20/value

    # Lire l etat du bouton poussoir
    z="$(cat /sys/class/gpio/gpio16/value)"

    # Boucler tant que le bouton poussoir n'est pas appuyé
    while [ $z -eq 0 ]
    # Le bloc inclus entre do et done est exécuté
    # par la boucle while

    # Début du bloc d instructions
    do
    # On lit l état du bouton poussoir
        z="$(cat /sys/class/gpio/gpio16/value)"
    # le bouton poussoir a été appuyé, on peut continuer
    done

    # Boucler tant que le bouton poussoir n'est pas relâché
    while [ $z  -eq 1 ]
    do
    # on lit l etat du bouton poussoir
        z="$(cat /sys/class/gpio/gpio16/value)"
    # le bouton poussoir est relâché, on peut continuer
    done

    # inverser l état de la LED (toggle)
    led=$((1-led))
```

```
# Fin du bloc d instructions
done
```