

Pour aller plus loin

Attribution - Partage dans les Mêmes Conditions :
<http://creativecommons.org/licenses/by-sa/3.0/fr/>

Table des matières

I - Exercice : Défis supplémentaires	3
Solutions des exercices	5

I Exercice : Défis supplémentaires

Suite de Fibonacci (1/2)

Créer un programme qui prends un nombre N en entrée et qui affiche les N premières valeurs de la suite de Fibonacci.

Suite de Fibonacci (1/2)

Remarque :

En bash, les variables sont considérée par défaut comme du texte. Pour additionner deux nombres, il faut faire :

```
1 resultat=$(( $nombre1 + $nombre2 ))
```

Question 1

[solution n°1 p. 5]

Écrire le script correspondant.

Suite de Fibonacci (2/2)

Créer un programme qui prends un nombre N en entrée et qui affiche les toutes les valeurs de la suite de Fibonacci inférieures ou égales à N.

Question 2

[solution n°2 p. 5]

Écrire le script correspondant.

Secret

Révéler un secret à l'utilisateur en fonction d'un mot de passe défini en variable d'environnement.

On doit vérifier que le mot de passe est bien défini sinon il y a a une erreur.

Secret

Remarque :

on peut passer une variable en paramètre à un exécutable en faisant :

```
1 NOM_VARIABLE=VALEUR commande
```

Question 3

[solution n°3 p. 6]

Écrire le script correspondant.

Jeu + ou -

Coder un jeu du + ou - . Générer un nombre aléatoire. L'utilisateur teste une valeur, le programme répond si la valeur à deviner est supérieure ou inférieure.

Bonus : permettre de choisir le maximum

Jeu + ou -

Remarque :

la variable RANDOM génère un nombre aléatoire,

L'opération modulo n permet d'obtenir le reste de la division entière par n

Question 4

[solution n°4 p. 6]

Écrire le script correspondant.

Solutions des exercices

Solution n°1

[exercice p. 3]

Suite de Fibonnaci (1/2)

```
1 #! /usr/bin/bash
2
3 echo "Entrer un rang n : "
4 read n
5
6 u0=0
7 u1=1
8
9 echo "u0 = $u0"
10 if [ $n -eq 0 ]
11 then
12 # on s'arrête ici si n = 0
13   exit 0
14 fi
15
16 echo "u1 = $u1"
17 if [ $n -eq 1 ]
18 then
19 # on s'arrête ici si n = 1
20   exit 0
21 fi
22
23 for i in $(seq 2 $n)
24 do
25     u2=$((u0 + u1))
26
27     echo "u$i = $u2"
28
29     u0=$u1
30     u1=$u2
31 done
```

Solution n°2

[exercice p. 3]

Suite de Fibonnaci (1/2)

```
1 #! /usr/bin/bash
2
3 echo "Entrer un nombre n : "
4 read n
5
6 u0=0
7 u1=1
8
9 echo "u0 = $u0"
10 if [ $n -eq 0 ]
11 then
12 # on s'arrête ici si n = 0
13   exit 0
14 fi
15
```

```

16 echo "u1 = $u1"
17 if [ $n -eq 1 ]
18 then
19 # on s'arrête ici si n = 1
20 exit 0
21 fi
22 i=1
23 while [ $u2 -le $n ]
24 do
25     echo "u$i = $u2"
26     i=$(( i + 1 ))
27     u2=$((u0 + $u1))
28
29     u0=$u1
30     u1=$u2
31 done

```

Solution n°3

[exercice p. 3]

Secret

```

1 #! /usr/bin/bash
2
3 if [ -z "$MOT_DE_PASSE" ]
4 then
5     echo "Aucun mot de passe n'est défini"
6     exit 1
7 fi
8
9 read PWD_DONNE
10
11 if [ "$PWD_DONNE" = "$MOT_DE_PASSE" ]
12 then
13     echo "Bravo tu as trouvé"
14     exit 0
15 else
16     echo "Tu n'as pas trouvé"
17     exit 1
18 fi

```

Solution n°4

[exercice p. 4]

Jeu + ou -

```

1 echo "Quel est le maximum ?"
2 read MAX_NUMBER
3
4 GENERATED_NUMBER=$((RANDOM % (MAX_NUMBER + 1)))
5 GIVEN_NUMBER=-1
6
7 while [ "$GIVEN_NUMBER" -ne "$GENERATED_NUMBER" ]
8 do
9     echo "Vous devez trouver un nombre entre 0 et $MAX_NUMBER"
10    echo "Quelle est votre réponse ?"
11    read GIVEN_NUMBER
12    if [ "$GIVEN_NUMBER" -lt "$GENERATED_NUMBER" ]
13    then
14        echo "C'est plus !"
15    elif [ "$GIVEN_NUMBER" -gt "$GENERATED_NUMBER" ]
16    then

```

```
17     echo "C'est moins !"  
18 else  
19     echo "Vous avez gagné !"  
20 fi  
21 done
```

