

# Exercices autour de la boucle « Tant Que »

## I. Évaluer un algorithme

Voici un algorithme écrit avec Algobox :

```

VARIABLES
├── x EST_DU_TYPE NOMBRE
├── y EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
├── x PREND_LA_VALEUR 0
├── TANT_QUE (x<=1) FAIRE
│   ├── DEBUT_TANT_QUE
│   ├── y PREND_LA_VALEUR 2*po w(x,2)-3*x+1
│   ├── AFFICHER y
│   ├── x PREND_LA_VALEUR x+0.1
│   └── FIN_TANT_QUE
└── FIN_ALGORITHME
    
```

- Exécuter à la main cet algorithme en complétant le tableau suivant dans lequel on indiquera pour chacune des étapes la valeur prise par les différentes variables :

	x	y
Étape 1		
Étape 2		
⋮		
⋮		

- Que fait cet algorithme ?
- Aurait-on pu obtenir le même résultat avec un autre algorithme ?

## II. Lancer d'un dé

- Voici un algorithme écrit avec Algobox :

```

VARIABLES
├── D EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
├── D PREND_LA_VALEUR floor(6*random()+1)
├── AFFICHER D
└── FIN_ALGORITHME
    
```

Que fait l'instruction :  $\text{floor}(6*\text{random}()+1)$  ?

- Ouvrir le fichier **ex\_2.alg** et tester cet algorithme.

- Voici un nouvel algorithme écrit avec Algobox :

```

VARIABLES
├── D EST_DU_TYPE NOMBRE
├── k EST_DU_TYPE NOMBRE
├── S EST_DU_TYPE NOMBRE
DEBUT_ALGORITHME
├── S PREND_LA_VALEUR 0
├── POUR k ALLANT_DE 1 A 10
│   ├── DEBUT_POUR
│   ├── D PREND_LA_VALEUR floor(6*random()+1)
│   ├── S PREND_LA_VALEUR S+D
│   ├── AFFICHER D
│   └── FIN_POUR
├── AFFICHER S
└── FIN_ALGORITHME
    
```

Que représente la variable S ?

- Ouvrir le fichier **ex\_2b.alg** et tester cet algorithme ; cela conforte-t-il la réponse donnée à la question précédente ?
- Élaborer un algorithme utilisant la boucle « Tant...Que » qui affiche le nombre de lancers d'un dé (cubique bien équilibré dont les faces sont numérotées de 1 à 6) nécessaires pour obtenir une somme supérieure ou égale à 13.  
Le programmer à l'aide d'Algobox et le tester.

## III. Puissances de 2

- Élaborer un algorithme permettant de déterminer la première puissance de 2 dépassant  $10^{99}$ .
- Le programmer à l'aide d'Algobox et le tester.