



# **CONCOURS D'ENTRÉE**

## **CYCLE INGENIEUR**

### **OPTION : INFORMATIQUE**

**Samedi 16 Avril 2016**

**Durée : 2 Heures**

## A) Liste vers 9...

### Spécifications :

- La fonction *liste\_vers\_9* (*entier AB*) affiche les éléments de la liste vers 9 générée et retourne la longueur (nombre d'éléments) de cette liste.

### Principe :

On teste si AB est compris entre 10 et 99 et si les deux chiffres qui le composent sont différents ( $AB \bmod 11 \neq 0$ ). Si ce n'est pas le cas, on affiche rien et on retourne 0.

Dans le cas contraire et ce jusqu'à ce que  $AB=9$ , on affiche AB, puis on inverse les deux chiffres de AB que l'on soustrait à AB pour obtenir la nouvelle valeur de AB. On augmente alors la longueur de la liste de 1, longueur initialisée à 1.

Enfin on affiche la longueur de la liste.

```
algorithme fonction liste_vers_9 : entier
parametres locaux
    entier AB
variables
    entier long /* la longueur de la liste */
debut
    si (AB < 10) ou (AB > 99) ou (AB mod 11 = 0) alors
        retourne (0)
    sinon
        ecrire (AB, ", ")
        long <- 1
        faire
            AB <- AB - 10 * AB mod 10 + AB div 10
            si AB < 0 alors
                AB <- -AB
            fin si
            ecrire (AB, ", ")
            long <- long + 1
        tant que AB <> 9
        ecrire ("il y a ", long, " éléments.")
        retourne (long)
    fin si
fin algorithme fonction liste_vers_9
```

## B) Arbres binaires et occurrences ...

Soit l'arbre  $B$  défini, sous forme d'occurrences et représenté à l'aide d'une chaîne de caractères de la manière suivante:

$$B = "E,0,1,00,01,10,11,001,101,1010,1011"$$

1) Sans représenter l'arbre  $B$ , pour déterminer :

- $T(B)$ , il suffit de compter les virgules présentes dans la chaîne et d'y ajouter 1
- $H(B)$ , il suffit de compter le nombre de 0 et de 1 de la dernière occurrence de la chaîne ((1011) dans l'exemple)

- c) Qu'un nœud n'est pas une feuille, il faut que son occurrence n'en préfixe pas une autre. Il suffit donc de parcourir le reste de la chaîne en vérifiant que l'occurrence de ce nœud ne se trouve pas au début des occurrences suivantes.
- d)  $LC(B)$ , il suffit de compter le nombre total de 0 et de 1 se trouvant dans la chaîne.
- e)  $HM(B)$ , il suffit simplement d'effectuer l'opération.

2) La représentation de l'arbre est la suivante :

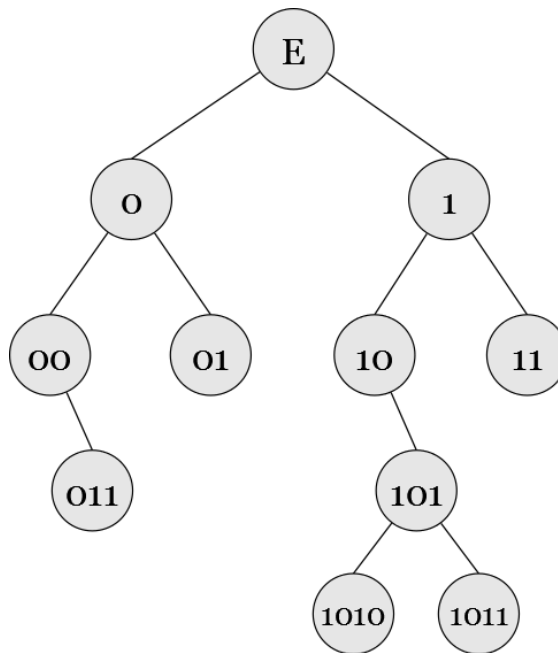


Fig. 1 – Arbre B

### C) Chemin impossible ...

Les séquences 2 et 4 sont impossibles :

- 1) S, on descend à gauche - A, on descend à droite - Q on descend à gauche - C, on descend à droite - M, on descend à gauche - **G**
- 2) Q, on descend à gauche - A, on descend à droite - K, on descend à gauche - C, on descend à droite - **O ne peut se trouver là, il n'est pas inférieur à K !**
- 3) A, on descend à droite - C, on descend à droite - K, on descend à gauche - I, on descend à gauche - E, on descend à droite - **G**
- 4) C, on descend à droite - K, on descend à gauche - I, on descend à gauche - **A n'est pas supérieur à C**

## D) Arbres de Léonard ...

- a. L'arbre  $F_6$  de Fibonacci est celui de la figure 2 dont les noeuds contiennent leur propre valeur de déséquilibre.

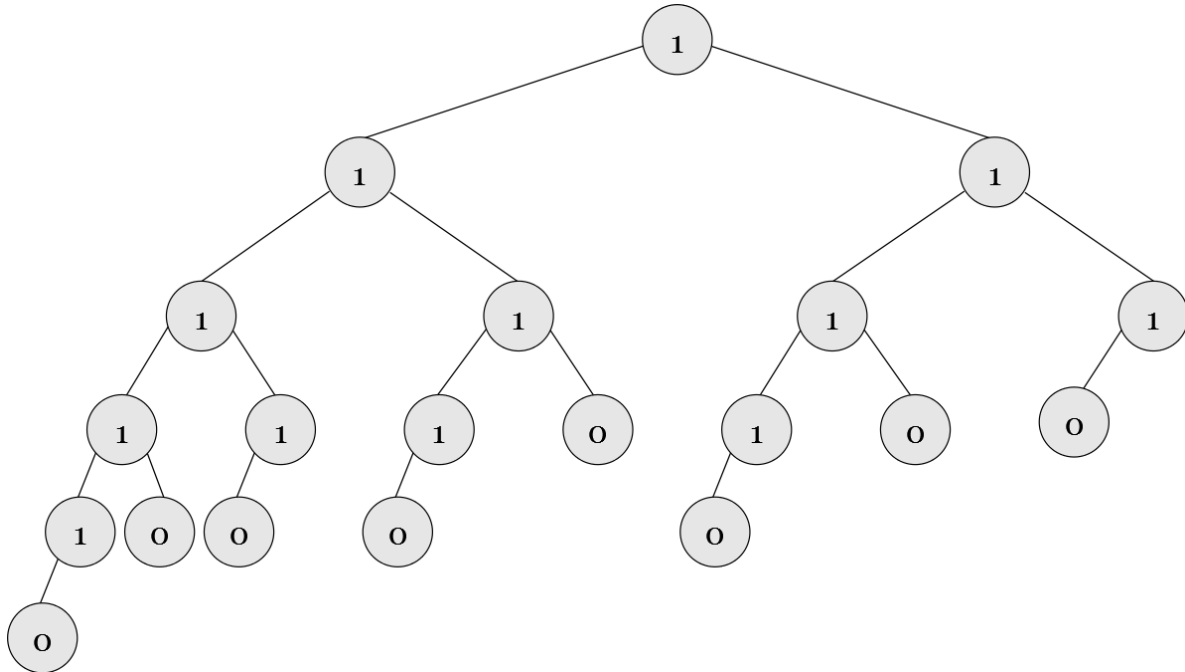


Fig. 2 – Arbre de Fibonacci  $F_6$

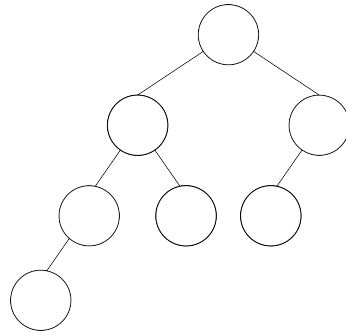
- b. Tableau des valeurs :

$n$	$H_n$	$T_n$	$F_n$	$Fib_n$
<b>0</b>	—	0	0	0
<b>1</b>	0	1	1	1
<b>2</b>	1	2	1	1
<b>3</b>	2	4	2	2
<b>4</b>	3	7	3	3
<b>5</b>	4	12	5	5
<b>6</b>	5	20	8	8

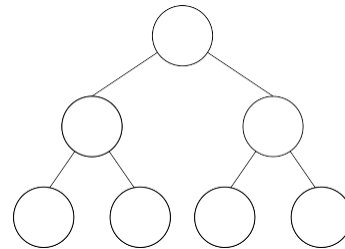
- c.  $H_n = n - 1$  évident par récurrence

$T_n = \text{Fib}_{n+2} - 1$  par récurrence aussi en notant que  $T_n = T_{n-1} + T_{n-2} + 1$  et en sachant que  $\text{Fib}_n = \text{Fib}_{n-1} + \text{Fib}_{n-2}$   
 $F_n = \text{Fib}_n$  Vérifient la même récurrence, donc  $F_n = \text{Fib}_n = \text{Fib}_{n-1} + \text{Fib}_{n-2}$

- d. Oui, il existe des arbres (1-Equilibrés) qui pour le même nombre de nœuds qu'un arbre de Fibonacci de hauteur  $h$  ont une longueur de cheminement externe supérieure à celui-ci.
- e. Comme le montre les figures suivantes, il existe au moins l'arbre parfait (1-Equilibrés) de 7 sommets dont la longueur de cheminement externe est supérieure à celle de l'arbre de Fibonacci de 7 sommets.



Arbre de Fibonacci à 7 nœuds (LCE=7)



Arbre parfait à 7 nœuds (LCE=8)

- f. Il y a deux façons de déterminer la longueur de cheminement externe d'un arbre de Fibonacci :
- la première (simple) consiste à faire le parcours en profondeur de l'arbre (arbre binaire) en maintenant la profondeur courante et la longueur de cheminement externe dans des paramètres globaux. A chaque fois que l'on rencontre une feuille, on accumule sa profondeur à la longueur de cheminement externe.
  - la deuxième est de faire la aussi un parcours en profondeur de l'arbre, mais en appliquant, avec  $g(x)$ ,  $d(x)$  et  $nbf(x)$  respectivement les fils gauche, fils droit et nombre de feuille de  $x$ , la formule de calcul de  $LCE(x)$  suivante :

$$\begin{cases} LCE(x) = 0 \text{ si } x \text{ est une feuille,} \\ LCE(x) = LCE(g(x)) + nbf(g(x)) + LCE(d(x)) + nbf(d(x)) \text{ sinon.} \end{cases}$$

Question1 : Trouver l'intrus :

- 1- Vsphere
- 2- VmWare
- 3- MPLS \*\*\*\*\*
- 4- VBox

Question2 : Trouver l'intrus :

- 1- java
- 2- python
- 3- androïd\*\*\*\*\*
- 4- caml

Question3 : trouver l'intrus:

- 1- MVC
- 2- assembleur \*\*\*\*\*
- 3- agile
- 4- scrum

Question4 : trouver l'intrus:

- 1- windows-sphone
- 2- ietf \*\*\*\*
- 3- androïd
- 4- ios

question5 : Quel est le serveur qui convertit un nom de domaine en une adresse internet:

- 1- dns \*\*\*\*
- 2- ftp
- 3- ssh
- 4- dhcp

question6 : Quel est l'outil de manipulation de bases de données:

- 1- nagios
- 2- linux
- 3- ipV6
- 4- posgres\*\*\*\*\*

question7 : Trouvez l'intrus parmi ces protocoles de gestion de domaine:

- 1- active directory
- 2- san\*\*\*\*
- 3- ldap
- 3- kerberos

Question 8: A quelle valeur hexadécimale correspond cette suite binaire:

1111 1110

- 1- 1A
- 2- FE \*\*\*\*\*
- 3- 1B
- 4- 41

Question 9: Un site sécurisé sur Internet utilise le protocole:

- 1- http
- 2- ftp
- 3- telnet
- 4- https\*\*\*\*\*

Question 10: Trouver l'intrus parmi ces outils de sécurité:

- 1- firewall
- 2- proxy
- 3- scrum \*\*\*\*\*
- 4- acl

Question 11: Parmi les outils suivants lequel ne supporte pas les VPN :

- 1- PPTP
- 2- MPLS
- 3- IPsec
- 4- RTC \*\*\*\*\*

Question12 :        S = 0

Début

S = S + 10

Imprimer S

Si S > 30 aller vers FIN

    Sinon aller vers Début

FIN



Stop.

Qu'imprime ce programme ?

- 1- 10 20 30 40 \*\*\*\*\*
- 2- 0 1 3 4
- 3- 20 30 40 50
- 4- 10 20 30

Question13 : I = 5

S = 0

Début

S = S + I

Si S > 0 aller à début

Fin

Imprimer S

Quel est le résultat de ce programme ?

- 1- Imprime 5
- 2- Imprime 0
- 3- Imprime 10
- 4- Il boucle\*\*\*\*\*

Question14 : For i = 1 to 2

Print\_string « bienvenue » ;

Print\_newline() ;

Done

Ce programme imprime :

- 1- bienvenue
- 2- bienvenue  
bienvenue\*\*\*\*\*
- 3- bienvenue bienvenue bienvenue
- 4- bienvenue bienvenue

Question15 : Que retourne l'instruction list(range(1,5)) ?

- 1- [1,2,3,4]\*\*\*\*\*
- 2- 1
- 3- 5
- 4- 3

Question16 : Qu'affichera : `print("python"[2])`

- 1- 'y'
- 2- 't'\*\*\*\*\*
- 3 -"py"
- 4- une erreur

Question17 : Qu'affichera : `print("12"+"34")`

- 1- 1234\*\*\*\*\*
- 2- 12,34
- 3- 46
- 4-Une erreur

Question18 : Qu'affichera : `print("python"[2:5])`

- 1- ython
- 2- ytho
- 3- tho\*\*\*\*\*
- 4-Une erreur

Question19 : Laquelle de ces expressions est correcte ?

- 1- `let a = 1 in let b = a + 1 ;;`
- 2- `let x = 1 in 0 < x < 1 ;;`
- 3- `let x = 1 in 2x + x^2 ;;`
- 4- `let a = 1 and b = 3 in let x = a + b and y = a - b in x * y ;;*****`

Question20 : Soit la fonction f définie ainsi : `let f x = x ;;`

Quel est le type de x ?

- 1- Entier
- 2- Flottant
- 3- Booléen
- 4- x pourrait être de n'importe quel type\*\*\*\*\*

Question21 : Parmi les affirmations suivantes, laquelle est fausse ?

- 1- Une liste peut contenir une liste
- 2- Caml renverra une erreur de type « Warning » lors de l'appel d'une fonction anonyme\*\*\*\*\*
- 3- Le type caractère existe en Caml
- 4- L'opérateur ^ permet de concaténer 2 strings

Question22 : Soit la fonction f définie ainsi, avec  $x > 0$  :

```
let rec f x =  
  if x < 10 then  
    1  
  else
```

1 + f (x/10) ;;

- 1- f calcule x+x
- 2- f renverra toujours la même valeur : x
- 3- f calcule le nombre de chiffres de x\*\*\*\*\*
- 4- f ne calcule rien, elle est incorrecte

Question23 : Soit la fonction f définie ainsi :

```
let f x y =  
  if y < 0 then  
    invalid_arg "j'aime pas les 0"  
  else  
    let rec g a b =  
      if b = 0 then 1  
      else  
        a * g a (b - 1)  
    in  
      g x y ;;
```

- 1- f n'est pas correcte
- 2- f retourne x \* y
- 3- f retourne x + x \* y si y > 0
- 4- f calcule x à la puissance y si y > 0\*\*\*\*\*

Question24 : Soit la fonction f définie ainsi :

```
let rec f = function  
  | [] -> 1  
  | e::l -> e * f l;;
```

- 1- f calcul le produit des éléments d'une liste\*\*\*\*
- 2- f retourne le dernier élément d'une liste
- 3- f retourne le premier élément d'une liste
- 4- f n'est pas correct

Question25 : Soit la fonction f définie ainsi :

```
let rec f = function  
  | [] -> failwith "la liste est vide"  
  | e::[] -> e  
  | e1::e2::l when e1 < e2 -> f (e1::l)  
  | e1::e2::l -> f (e2::l);;
```

- 1- f trie une liste en ordre croissant
- 2- f retourne la valeur minimale d'une liste\*\*\*\*\*
- 3- f retourne le dernier élément d'une liste
- 4- f ne s'arrête jamais