

ALSDS : ATELIER 1

1. RECOMMANDATIONS

1. Lisez attentivement votre sujet, soignez votre travail, faites le maximum de choses mais bien !
2. Construisez vos modules un à un, mais laissez-les sous forme de modules internes dans votre programme principal à l'exception des modules déjà existants qui doivent être externes.
3. Votre solution doit **ABSOLUMENT** tenir compte du formalisme étudié en cours et utiliser la modularité !

ALSDS : ATELIER 1

2. ENONCÉ

Nous disposons d'un tableau X à une dimension et contenant des nombres entiers dont certains appartiennent à la suite de Fibonacci (1, 1,2, 3, 5,...) , d'autres ont une partie centrale composée des mêmes chiffres (exemples : 1722259, 56777734) et des intrus, qui n'appartiennent à aucune des deux catégories précédemment citées.

Travail à Faire :

Partie A (à traiter complètement) :

Nous souhaitons construire le tableau (R) à deux dimensions de la façon suivante.

La 1^{ère} ligne contiendra tous les éléments de X appartenant à la suite de fibonacci

La 2^{ième} ligne contiendra tous les éléments dont la partie centrale est composée des mêmes chiffres

La 3^{ième} ligne contiendra tous les intrus!

ALSDS : ATELIER 1

3. CONCEPTION

Découpage modulaire : Il faut :

- lire la table initiale qui contient les nombres donnés (module **Lect1D**)
- trouver les nombres qui appartiennent à la suite de Fibonacci (module **ElemFibo**)
- trouver les nombres qui ont une partie centrale composée des mêmes chiffres (module **PartCent**)
- trier des lignes (module **Tri_Sel** ou n'importe quel autre module de tri disponible)
- le remplissage du tableau R , étant particulier, il se fera dans l'algorithme principal
- afficher le tableau R à 2 dimensions (Module **Ecrire2D**)

ALSDS : ATELIER 1

3. CONCEPTION

Découpage modulaire : (suite)

Pour construire ElemFibo , il faut :

- trouver le Nième élément de la suite de Fibonacci (module **Fibo**)
- disposer des 45 éléments de la suite de Fibonacci en les mettant dans une table (module **SuitFibo**). Le module Fibo est déjà traité (série2), donc il faut construire le module **SuitFibo**.

Fibo(n :ntier) : longint // donne le nième élément de la suite de Fibonacci

ALSDS : ATELIER 1

3. CONCEPTION

Découpage modulaire : (suite)

Pour construire le module PartCent, il faut :

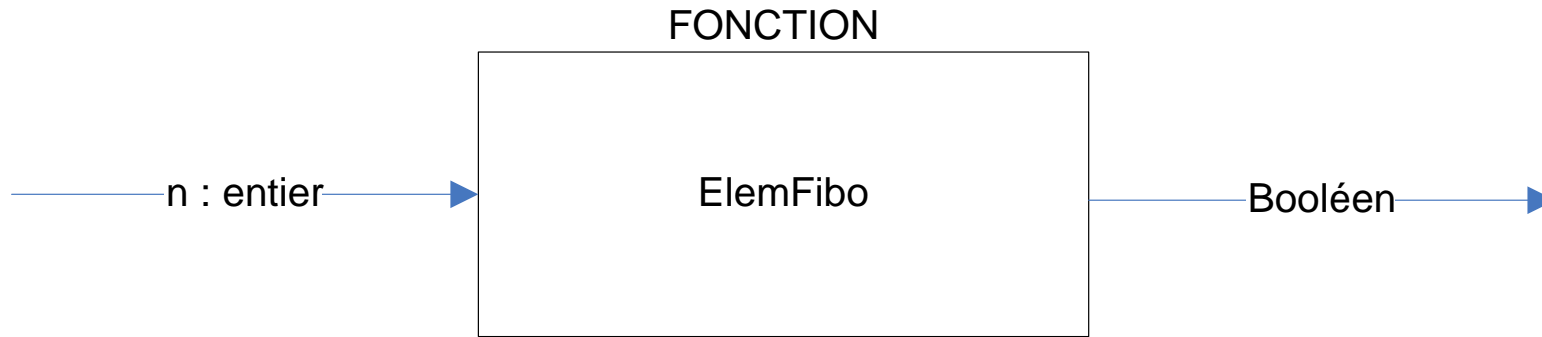
- connaître le nombre de positions d'un nombre N (module NbPos)
- savoir comment extraire des positions d'un nombre (module Extr_Nb)

NbPos(n : longint) : longint // donne le nombre de positions de N

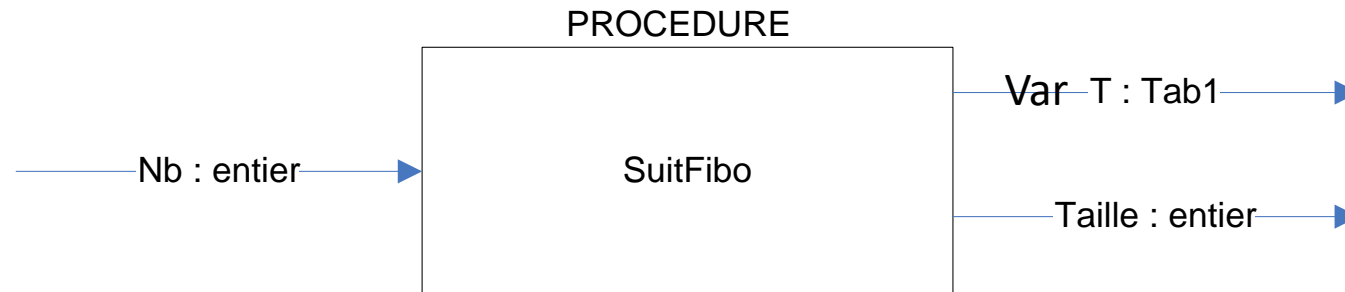
*Extr_nb(n, c, p : longint) : longint // extrait de N un nombre de c positions à
partie de la position p (incluse)*

ALSDS : ATELIER 1

3. CONCEPTION



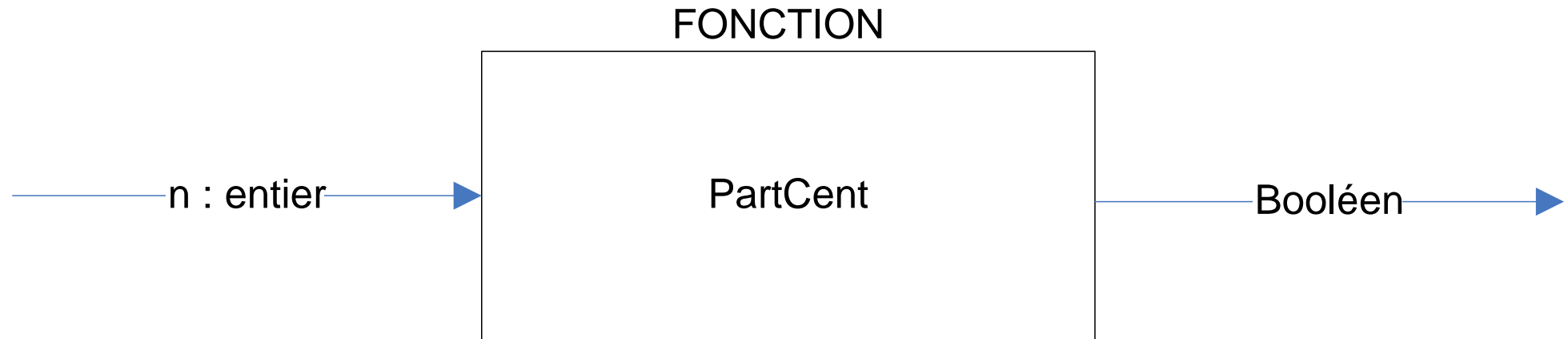
Rôle : donne vrai si N se trouve parmi les 45 premiers éléments de la suite de Fibonacci



Rôle : donne une table qui contient les éléments de la suite de Fibonacci
Attention Nb = 45 au maximum

ALSDS : ATELIER 1

3. CONCEPTION



Rôle : donne vrai si la partie centrale de N est composée du même chiffre

ALSDS : ATELIER 1

3. CONCEPTION

a./ Construction du module **SuitFibo** :

Idée générale : on met dans la table le 1^{er} élément de la suite, puis le second, puis le troisième.... On ne dépasse pas le 45^{ième} pour la simple raison que le type Longint ne suffit plus.

a.1/ Analyse

- on initialise T[1] et T[2] à 1
- on fait varier $i = 3, 4, 5, \dots, 45$
 - $T[i] = \text{Fibo}[i]$ (on met dans le tableau l'élément de rang i de la suite de Fibonacci)
- $\text{tail} = N$

ALSDS : ATELIER 1

3. CONCEPTION

a.2/ Algorithme

Procedure **SuitFibo** (N : entier ;var T : tab1D; var Tail : entier)

variable i: entier

Fonction fibo

..... Corps de Fibo

DEBUT

T[1] \leftarrow 1

T[2] \leftarrow 1

Pour i allant de 3 à n faire t[i] \leftarrow Fibo(i)

Tail \leftarrow N

FIN

ALSDS : ATELIER 1

3. CONCEPTION

b./ Construction du module **ElemFibo** :

Idée générale : si l'élément donné se trouve dans la table qui contient les éléments de la suite, alors il appartient à la suite de Fibonacci

b.1/ Analyse

- on construit la table qui contient les 45 premiers éléments de la suite de Fibonacci : **SuitFibo(45, t, tai)**
- Trouv = Faux (*elemfibo = Faux: on prend un aiguillage qui prendra la valeur faux au départ*)
- on parcourt la table avec $i : 1, 2, 3 \dots$
 - Si $T[i] = N$, on met vrai dans Trouv (*elemfibo = Vrai*)
 - On s'arrête lorsque (Trouv= vrai) ou ($i > Tai$)

ALSDS : ATELIER 1

3. CONCEPTION

b.2/ Algorithme

Function **ElemFibo**(n:entier): booléen

Variables T : tab1D; i , Tai : Entier; Trouv :Booléen

Peocédure SuitFibo

... Corps de la procédure ...

DEBUT

SuitFibo (45, T, Tai)

Trouv ← Faux

i ← 1

Répéter

Si T[i] = n Alors Trouv ← Vrai

i ← i+1

Jusqu'à (Trouv= Vrai) OU (i>tai)

ElemFibo ← Trouv

FIN

ALSDS : ATELIER 1

3. CONCEPTION

c./ Construction du module **PartCent** :

Idée générale :

- Si le nombre de positions de N est pair, il suffit que les 2 chiffres du milieu soient identiques pour dire que la partie centrale de N est composée des mêmes chiffres.
- De la même manière si le nombre de positions est impair, on ne contrôlera que les 3 chiffres du milieu.
- Pour vérifier si les chiffres du milieu sont identiques on pourra soit , utiliser le module MemeCh, soit vérifier si les 2 ou 3 chiffres du milieu sont égaux à 0 , soient s'ils sont des multiples de 11 ou de 111 (par exemple dans le cas où les 2 chiffres du milieu sont identiques on ne peut avoir que 11,22,33,44,....,99).
- Aussi on évitera les nombres composés de 2 et de 3 chiffres parce que l'on ne peut en extraire les parties centrales.

ALSDS : ATELIER 1

3. CONCEPTION

c./ Construction du module **PartCent** :

c.1/ Analyse

- partCent = *Faux* (on prend un aiguillage initialisé à faux au départ)
- Si le nombre de positions est paire et supérieur à 2
 - On extrait le nombre composé des 2 chiffres du milieu
 - Si ce nombre est égal à 0 ou il est divisible par 11 alors PartCent = vrai

Sinon

- Si le nombre de positions est supérieur à 2
 - On extrait le nombre composé des 3 chiffres du milieu
 - Si ce nombre est égal à 0 ou il est divisible par 111 alors PartCent = vrai

ALSDS : ATELIER 1

3. CONCEPTION

c.1/ Algorithmme

Fonction **PartCent** (n : entier) : Booléen

Variables Nbext : entier

Fonctions Extr_Nb , NbPos

DEBUT

partCent \leftarrow Faux

SI (NbPos(n) mod 2 = 0) ET (NbPos(n) > 2)

Alors

Dsi

Nbext \leftarrow extr_Nb(n, 2, NbPos(n) div 2)

SI (Nbext = 0) or (Nbext mod 11 = 0) Alors

PartCent \leftarrow Vrai

Fsi

Sinon

SI NbPos(n) > 3 Alors

Dsi

Nbext \leftarrow extr_Nb(n, 3, NbPos(n) div 2)

SI (Nbext = 0) OU (Nbext mod 111 = 0)

Alors PartCent \leftarrow Vrai

Fsi

FIN

ALSDS : ATELIER 1

3. CONCEPTION

e./ Construction de l'algorithme principal

Idée générale : comme il demandé que les lignes du tableau à 2 dimensions soient triées il suffit pour cela de trier au départ le tableau donné X . Puis on prend les éléments du tableau trié X, un à un on va voir :

- Si cet élément appartient à la suite de Fibonacci on le met dans la 1^{ère} ligne de R
- S'il a une partie centrale composée des mêmes chiffres on le met dans la 2^{ième} ligne de X
- S'il n'est ni l'un , ni l'autre on le met dans la 3^{ième} ligne

A la fin on affiche R . Pour avoir le format désiré, il suffit de reprendre le module Ecrire2D , et de le modifier un peu.

Nota : pour éviter d'avoir des lignes trop longues avec des colonnes qui ne contiennent que des zéros. On affichera que les colonnes qui contiennent effectivement des éléments recherchés.

ALSDS : ATELIER 1

3. CONCEPTION

f.1/ Analyse

- On lit le tableau X : **Lect1D(X,tai)**
- On le trie **Tri_Sel(X, Tai) : Tri_sel(X, Tai)**
- On initialise i à 0 (*i est l'indice de la 1^{ère} ligne qui va contenir les éléments appartenant à la suite*)
- On initialise j à 0 (*j est l'indice de la 2^{ième} ligne des éléments dont la partie centrale et identique*)
- On initialise k à 0 (*k est l'indice de la 3^{ième} ligne qui va contenir les intrus*)
- On fait varier m : 1, 2, 3, ..., Tai et à chaque fois (*pour prendre un à un les éléments du tableau X*)

- SI X[m] appartient à la suite de Fibonacci
 - On incrémente i
 - On met X[m] dans R[1,i]
- SI X[m] a une partie centrale composée des mêmes chiffres
 - On incrémente j
 - On met X[m] dans R[2, j]
- SI X[m] n'appartient à aucune des 2 catégories précédentes
 - On incrémente k
 - On met X[m] dans R[3,k]
- On écrit R
- Ecrire2D (R,3,m)

ALSDS : ATELIER 1

3. CONCEPTION

f.2/ Algorithme

Constante max = 1000; maxL = 1000; maxC = 1000;

Type Tind = 1..max;
 TindL = 1..maxL; TindC = 1..maxC;
 Tab1D = Array[Tind] Of Longint;
 Tab2D = Array[TindL,TindC] Of Longint;

Variables

 X: Tab1D; R: Tab2D;
 Tai : Tind;
 i, j, k, m: entiers

Procédures Lect1D , Tri_sel , Ecrire2D

Fonctions PartCent , ElemFibo

ALSDS : ATELIER 1

3. CONCEPTION

DEBUT

Lect1D (X,Tai)

Tri_Sel (X,Tai)

$i \leftarrow 0 ; j \leftarrow 0 ; k \leftarrow 0 ;$

Trouv \leftarrow False

POUR m Allant de 1 à Tai Faire

DPOUR

SI ElemFibo (X[m]) Alors

DSI

$i \leftarrow i+1$

$R[1, i] \leftarrow X[m]$

Trouv \leftarrow Vrai

FSI

SI PartCent (X[m]) Alors

DSI

$j \leftarrow j+1$

$R[2, j] \leftarrow X[m]$

Trouv \leftarrow Vrai

FSI

SI (Trouv = False) Alors

DS

$k \leftarrow k+1$

$R[3, k] \leftarrow X[m]$

FSI

FPOUR

Ecrire2D (R, 3, m)

FIN

ALSDS : ATELIER 2

----- Atelier 1 -----

Tableau de donnees X

144| 479| 13| 1| 832040| 22222| 6765| 234478| 89| 4768|
15559| 4568|

Tableau de donnees X trie

1| 13| 89| 144| 479| 4568| 4768| 6765| 15559| 22222|
234478| 832040|

Tableau R (2D)

1| 13| 89| 144| 6765| 832040|
15559| 22222| 234478| 0| 0| 0|
479| 4568| 4768| 0| 0| 0|

ALSDS : ATELIER 1

4. ENONCE PARTIE 2

Partie B : Si nous disposons de plusieurs tableaux de la nature de X et on voudrait associer chacun d'eux à son tableau résultat (R) et les garder dans un enregistrement, Donnez la description de cet enregistrement.

Solution: Déclaration de l'enregistrement :

Type Tab1D = Tableau [1..1000] d'entiers
 Tab2D = Tableau [1..1000,1..1000] d'entiers
 E = **Enregistrement**
 X : Tab1D
 R : Tab2D
Fin
 Tab3 = Tableau [1..1000] de E

Fin de l'Atelier : 1CP1
Merci pour votre Attention

Réussite et Succès

