

INFORMATIQUE



avec F. BILLY, Olivier SAVET, Jean GALLICE

On va faire de la programmation!

On va utiliser Microsoft Visual Studio et Microsoft Excel YOUHO!

le maître d'ouvrage est interface entre le maître d'œuvre et l'utilisateur

le maître d'œuvre est celui qui réalise le produit.

le maître d'ouvrage ne sait pas faire, mais il sait ce qu'il est possible de faire, et il sait demande précisément ce qu'il veut.

framework est une bibliothèque d'outils sur lesquels on s'appuie pour implémenter (coder)

Au sé, on devra faire une maquette en VBA

CHAPITRE 1

1 Notion d'information

info : élément abstrait susceptible d'augmenter notre degré de connaissance d'un phénomène quelconque.
Pour être perçu, elle doit être codée et stockée sur un support!

exemple : info → ce que l'enseignant a à raconter
code → diaporama, discours
stockage → notes, mémoire de l'étudiant

* "Première machine à traiter l'info" : l'homme :
◦ réception → sens (vue, ouïe...)
◦ trait^{ement} → cerveau
◦ communicat^{ion} du résultat → écrit ou oral
◦ stockage → cerveau

informatique = information + automatique : traitement de l'info avec l'électronique

2 questions : 1. l'info existe-t-elle?
2. que veut-on en faire?

exemple : on cherche la liste des étudiants

1. l'info existe-t-elle? → 1^{er} sept → incomplète!

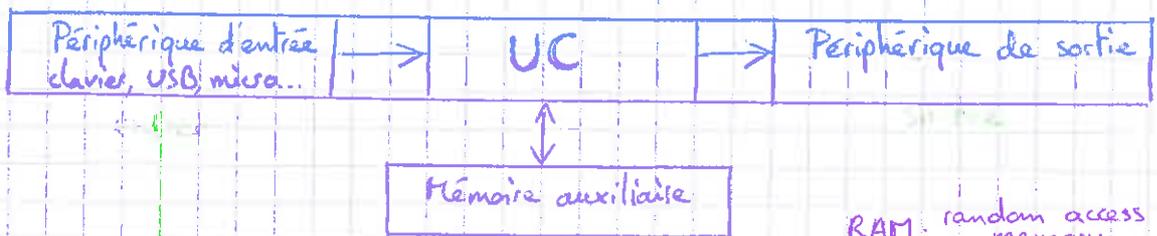
1^{er} oct → OK!

2. que veut-on en faire? → que veut-on comme info?
nom, prénom, adresse...

→ comment la communiquer?

mail, carte postale, annonce en amph

* Ordinateur



RAM: random access memory
HDD: hard drive disk (=disque dur)

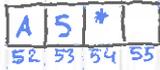
2 L'unité centrale

Principal composant: la carte mère avec un processeur (parfois ac un chipset) avec des bus (PCI, AGP...) avec la RAM avec des sorties

→ données graphiques
PCI-Express

On a aussi l'alim, la ventilation, le boîtier

La RAM est un genre d'ensemble de cases: dans chaque case, on peut stocker 1 caractère

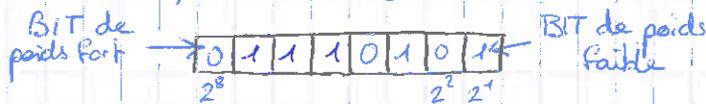


Les cases sont numérotées, chaque caractère a son adresse. Les expressions sont stockées dans un ensemble de cases successives identifiées par l'adresse de début et la longueur.

Dans l'UC, il n'existe que 2 états stables: - passage de courant (1)
- rien (0)

C'est le BIT (binary digit), la position élémentaire.

L'OCTET est combinaison de 8 BITS. C'est l'unité de base du stockage. Ça nous donne $2^8 \rightarrow 256$ combinaisons



1 octet stocke 1 caractère

Codification des caractères: le code ASCII est la norme internationale

exemples:

A	→	01000001	65	41
a	→	01100001	97	61
0	→	00110000	48	30
9	→	00111001	57	39
espace	→	00100000	32	20
+	→	00101011	43	2B
entrée	→	00001101	13	0D

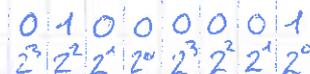
Calcul de la valeur en base 10 du A:

$$2^6 + 2^0 = 65$$



Calcul de la valeur en base 16 du A:

$$2^2 \text{ et } 2^0 \rightarrow 4 \text{ et } 1 \rightarrow 41$$



Base 10	Base 16
10	A
11	B
12	C
13	D
14	E
15	F
16	

Les caractères 0 à 31 et 127 ne sont pas affichables, ils correspondent à des commandes de contrôle de terminal informatique.

Le 32 est l'espace.

Les autres sont chiffres arabes et lettres latines (maj et min) et signe de ponctuation.

Pour stocker des valeurs numériques...

- entières → transformation base 10 - base 2 (2 octets)
- réelle → stockage en puissance de 10 (4 octets)

alors qu'un caractère tient en 1 octet

3 Le logiciel (Software)

OS = Operating System

C'est l'ensemble des programmes permettant l'utilisation de l'ordinateur.
Le matériel (Hardware) est accompagné d'un composant logiciel : le BIOS
Ensuite ya l'OS (système d'exploitation) comme Windows, Linux, Unix, Mac OS... Basic
Input
Output
System

Le OS gère tout :
- l'échange des données entre les composants
- la transmission des données saisies
- l'organisation du stockage des disques
- la liaison Homme - Machine par interface graphique

IHM = interface homme-machine

18/01/2012

→ Programmes d'applications (logiciels)

• réponse à un besoin spécifique (traitement de texte, tableur, JEU...)

• basé sur l'OS

• 2 types → standard (logiciel)

→ spécifique (pour une entreprise par exemple)

• ces logiciels sont réalisés avec → une plateforme de développement (langage de prog. BASIC, C, C++, VB...)
→ un tableau, base de données

Dans Visual Basic, on a des onglets pour :
→ l'IHM
→ Fenêtre code
→ éventuelle fenêtre(s) module

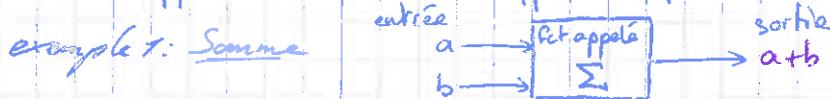
Fonctions & Procédures, Echanges de données

I Fonctions

Fonction \approx boîte noire (on ne connaît pas forcément le code...)

La fonction appelante ne connaît pas les détails de la fonction appelée

La fonction appelée est liée à l'appelante par des entrées/sorties (E/S)



Fonction = section de code produisant un résultat unique (sortie)
éventuellement avec une/des données(s) fournies.

⇒ définir une fct :
• 1 sortie unique ("return")
• établir les entrées des données (paramètres, ou arguments)
• établir le traitement de la fct

booléen = 2 résultats possibles

Syntaxe sur VB

Public * Function Nom de la fct (liste Paramètres) * As type Retour

Types : integer (entier)
string (chaîne de caractères)
boolean (booléen)
tab (d'entier, de caractère...)
double (réels)

* instructions *

Return Expression

End Function

* ça peut être "Private" mais on s'en pas pausiquoi

* listeParamètre c'est

ByVal
ou
ByRef Nom paramètre As type Paramètre

Le "pipe" : | c'est pour "ou bien" et "*" c'est pour "une ou plusieurs fois"

Exemple

```
Public Function SurfaceRectangle (ByVal a as Integer, ByVal b as Integer) As Integer  
    Return a * b  
End Function
```

Pour utiliser une fct, on l'appelle par son nom et on lui passe tous les paramètres requis.

exemple: (code-erreur =) MsgBox("rectangle de 2 par 3: " & SurfaceRectangle(2,3))

"SurfaceRectangle" : 2 paramètres
"MsgBox" : 1 type paramètre

Si on a codé une fct "est premier", on l'appel comme ça :

rep = est-premier(23) ici, "rep", boolean, prend la valeur "true"

Ring :

- si la fct n'a pas de paramètre, on met des parenthèses vides
- une fct peut appeler d'autres fct
- une fct répond à un besoin précis

En algorithmique, ça ressemble à ça :

```
resultat ← est-prem(n-tar)
```

Rmq: c'est le compilateur qui fait le lien entre les variables des fct appelante et appelées

II Procédures

Une procédure est une fct qui n'a pas de sortie, qui ne retourne rien.
Sur VB, ce sont les Subroutines : "Sub"

VB [Syntaxe : Public Sub NomProc (liste Paramètres)
End Sub ↘ comme pour fct

III Procédures événementielles

Procédure liée à un événement : clic, move, etc... Donc en g^{el} associé à l'HTML

VB [Public Sub NomProc (...) Handle Bouton Afficher.Clic
↳ méthode
↳ peut supprimer

IV Passage par valeur et par référence

- "ByVal" et "ByRef" décrivent comment une fct reçoit les paramètres de son appel
- ByVal → une copie de la valeur est transmise → la valeur ne change pas
- ByRef → la référence (adresse mémoire) est transmise → la valeur est transformée par la fct appelée

RMQ, FAQ

→ les tableaux sont des variables indicées

"Dim Tab() As Integer" : toutes les cases sont des entiers, tableau ∞

"Dim Tab(15) As Integer" : les cases de notre tableau de 16 cases sont des entiers

"Redim Tab(5)" : le tableau est défini à 6 cases et reseté

"Redim Preserve Tab(12)" : défini à 13 cases, mais garder les valeurs entrées.

structures de contrôles → séquence (sur visual basic = 1 ligne)
 → condition
 → boucle

IHM interface homme-machine

structures de données → constantes
 → variables
 → tableaux
 → collections

programmation événements ≠ séquentielle

Exemple et diapo

IDE : environnement de développ^{mt} intégré
 il contient → un éditeur de texte
 → compilateur(s)

Dans la zone label, on peut écrire :

Label1.Text = "Bonjour"

Sur visual - Nouveau projet → on prend le truc par défaut. Visual Basic sous windows
 → application windows forme → OK

Affichage → boîte à outils

Propriétés (de la fenêtre)

F5 → raccourci exécuter

Declares une variable : Dim "nom de la variable" As integer
 ici Dim (integer c'est un entier, mais on peut mettre autre chose)

Type la variable Div = 2
 Fonction modulo : Mod (ex = 2 Mod 14)

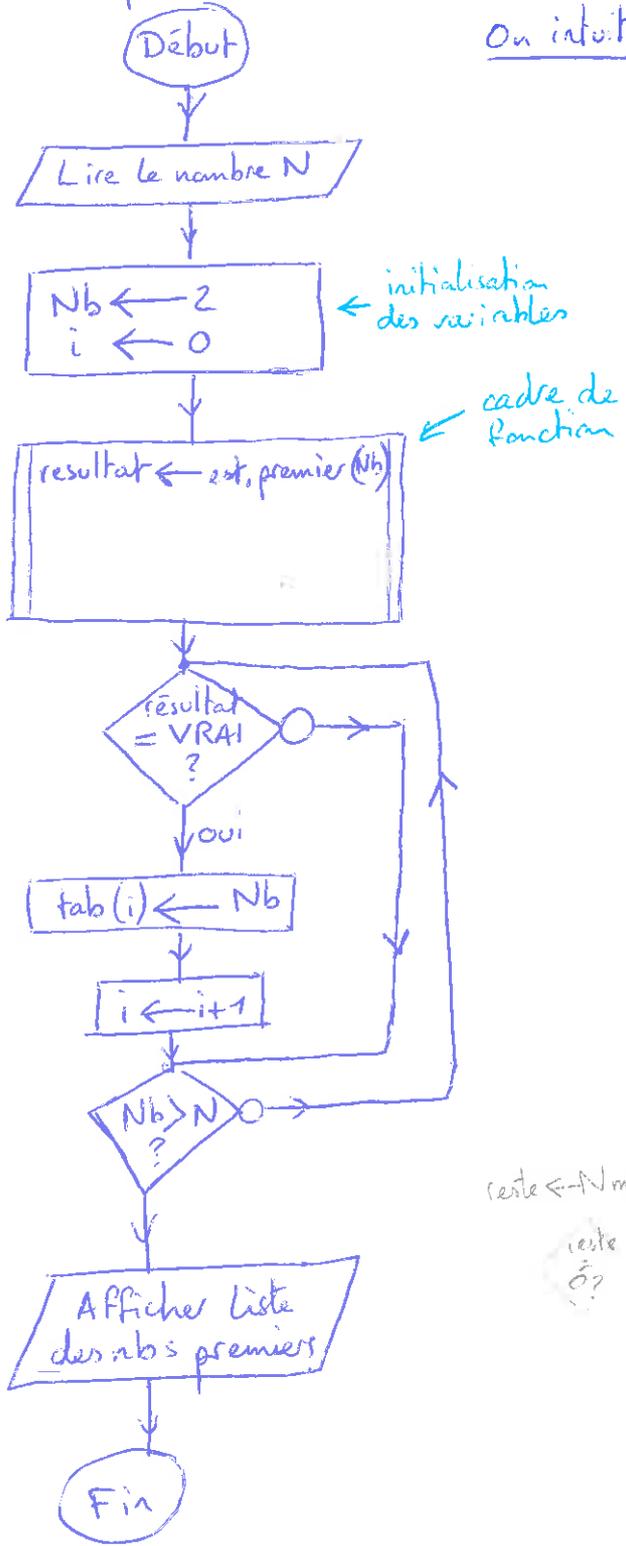
if <> = else and if
 if then, else and if
 : do until loop
 : Math.Sqrt(x)

répète
 sur

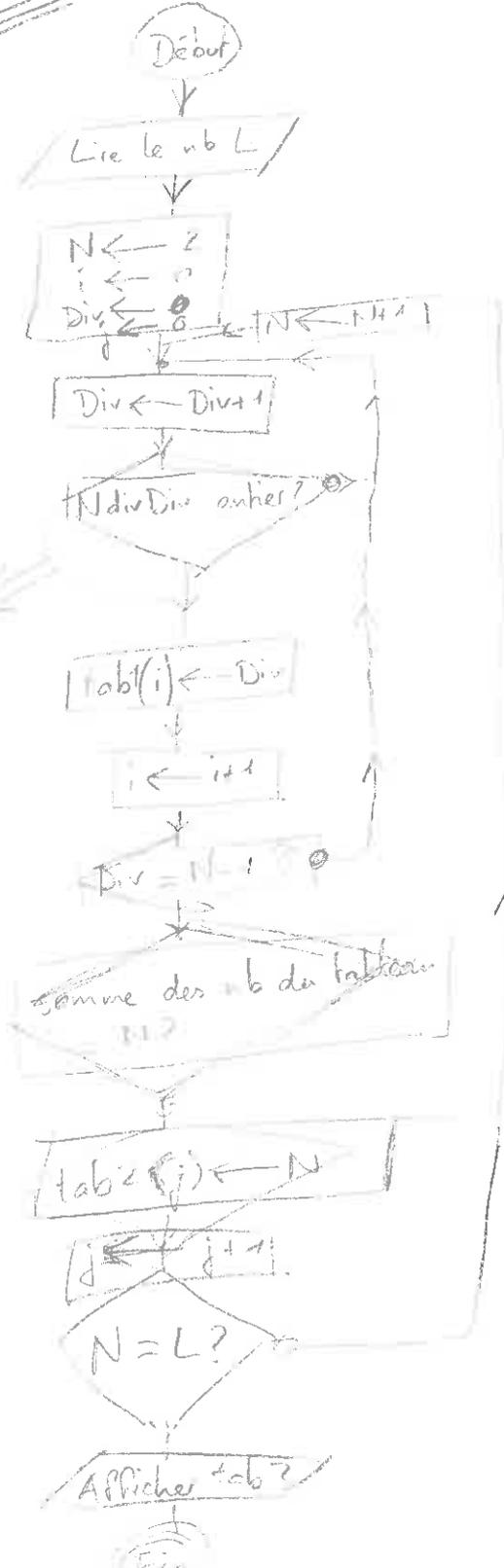
Autre possible écriture du pseudocode sur les listes de nb premiers

On initialise les variables

- N (entier) \rightarrow nb limite
- Nb (entier) \rightarrow nb en cours de test
- i (entier) \rightarrow indice du tableau
- tab (tableau d'entiers) \rightarrow tableau de stockage
- resultat (booléen)



TP4



INFORMATIQUE

Programmation

TP1 →

TP2 et 3 → Analyse d'un problème & premiers algorithmes

TP4 → Logique de programmation
ordi → Dossier Nb premiers

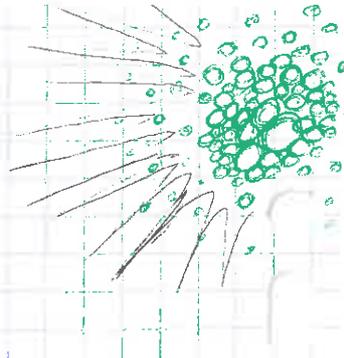
TP5 → Initiation aux macros Excel

—

TP6 → Macros Excel

TP7 → Initiation SGBD sous Access

Correction du pseudo code



△ Le tableau comprend en fait la case 0.

quand on ne sait pas cb de tps une boucle va tourner
↳ Répéter ...
Jusqu'à ce que

quand on sait cb de tps une boucle va tourner
↳ Pour I variant de X à Z, faire ...
Fin faire

↳ on peut définir le pas
ex: step 2, step -1

DEBUT PROCEDURE

ECRIRE "entrez une valeur limite"

LIRE N

$i \leftarrow 1$

Tab_NbPrem(i) $\leftarrow 2$

POUR Nb VARIANT DE 3 À N FAIRE

Div $\leftarrow 1$

REPETER

Div \leftarrow Div + 1

Reste \leftarrow Nb mod Div

JUSQUA CE QUE Reste = 0 OU Div $>$ $\sqrt{\text{Nb}}$

SI Reste $<>$ 0

ALORS $i \leftarrow i + 1$

Tab_NbPrem(i) \leftarrow Nb

FIN SI

FIN FAIRE

Liste \leftarrow ""

POUR J VARIANT DE 1 à i FAIRE

Liste \leftarrow Liste & Tab_NbPrem(j) & "-"

FIN FAIRE

ECRIRE Liste

FIN PROCEDURE

ISARA-Lyon DSI	TP n°2	Promotion 43
Informatique 2A F. BILLY / O. SAVET	Analyse d'un problème & premiers algorithmes	Septembre 2011

Objectifs des TPs n°2 & 3 :

A l'issue de ces séances, les étudiants doivent être capables de :

- Analyser l'énoncé d'un problème simple (à caractère mathématique).
- Appliquer (algorithmiquement, c'est-à-dire "sur feuille") les concepts suivants :
 - **Utiliser des variables** (variables isolées, compteurs et accumulateurs).
 - **Affecter** une variable (en mémoire).
 - **Lire** une donnée entrée au clavier.
 - **Écrire** un résultat (ou d'une donnée "intermédiaire") à l'écran.
 - **Effectuer des calculs** (traitement des données).
 - **Tester** la valeur d'une variable.
 - **Répéter** un traitement (notion de **boucle**).
 - Appliquer les **règles d'écriture des organigrammes** / algorigrammes.
 - Réaliser un organigramme.
 - Appliquer les **règles d'écriture du pseudo-code**.
 - Écrire un pseudo-code.

Exercice 1

Écrire un algorithme (organigramme, puis pseudo-code) permettant de trouver, puis d'afficher, la plus grande de 2 valeurs A et B données.

Exercice 2

Dans les années 2025, à l'ISARA, les étudiants de 1^{ère} année ont à leur programme 8 examens de mathématiques (les pauvres !).

L'étudiant Jean-Luc Martin a obtenu les notes suivantes :

10, 15, 2, 9.5, 9.5, 13, 5, 19

On souhaite calculer quelques paramètres statistiques concernant cet élève et ses notes de maths : **moyenne, variance et écart-type**.

Écrire un algorithme (organigramme, puis pseudo-code) renvoyant les trois paramètres statistiques ci-dessus.

On rappelle les formules associées à ces paramètres :

- moyenne :

$$m = \bar{x} = \frac{1}{N} \sum_{i=1}^N x_i$$

T.S.V.P. ⇨

- variance :

$$V = \left(\frac{1}{N} \sum_{i=1}^N x_i^2 \right) - \bar{x}^2$$

- écart-type :

$$\sigma = \sqrt{V}$$

NB : on demandera à l'utilisateur, après chaque donnée entrée, s'il a d'autres notes à saisir.

Exercice 3

Compléter l'exercice 2 pour obtenir, en sortie de l'algorithme, l'**étendue** des notes de Jean-Luc Martin.

On rappelle que l'**étendue** (encore appelée **amplitude**) d'une série de données est l'écart entre les valeurs extrêmes de la série.

NOTES informatique



Algorithmie : étude de la résolution d'un problème par la mise en œuvre d'une suite d'opérations | patate patate.

ya une histoire de CODES

langage de programmation de cette année → basic

On utilise des variables (numériques ou alphanumériques) stockées provisoirement en RAM

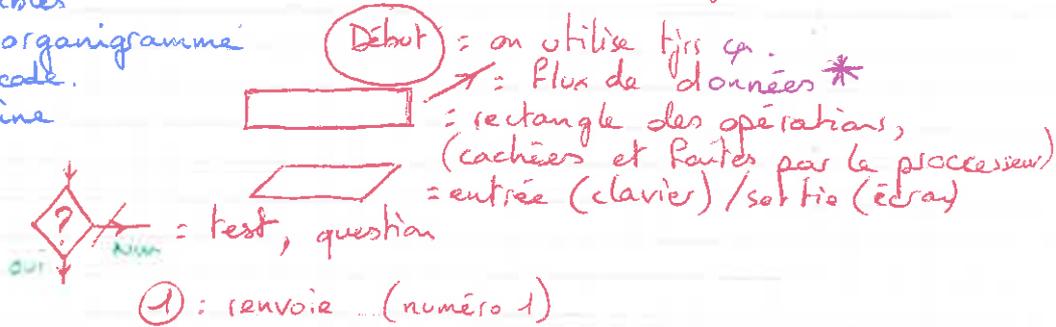
Opérateurs : + - * / ^ & ET OU

Ne pas confondre "←" avec "="
(affectation) (comparaison)

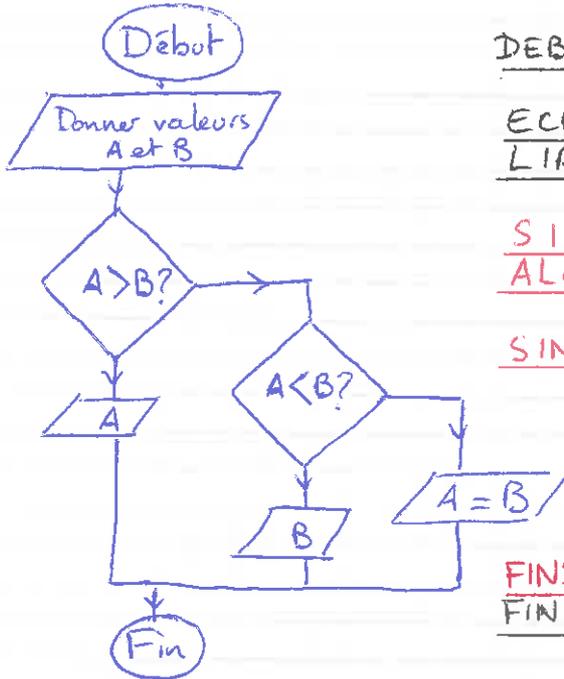
Résoudre un problème

- 1- On liste les variables
- 2- On réalise un organigramme
- 3- Ecrire le pseudo code.
- 4- Codage sur machine

Symboles d'organigramme



TPA Ex 1



DEBUT PROCEDURE

ECRIRE "Donnez la valeur de A") idem B
LIRE A

SI A > B
ALORS

SINON

ECRIRE "la valeur la + grde est A =", A

SI A < B

ALORS i

ECRIRE "la valeur la + grde est B =", B

SINON

ECRIRE "les deux valeurs sont égales"

FINSI

FINSI

FIN PROCEDURE

*NB = on entre tjrs par le haut, on ressort tjrs par le bas

Ex. 2

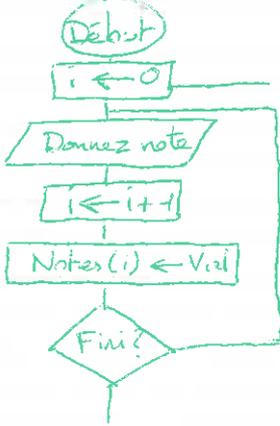
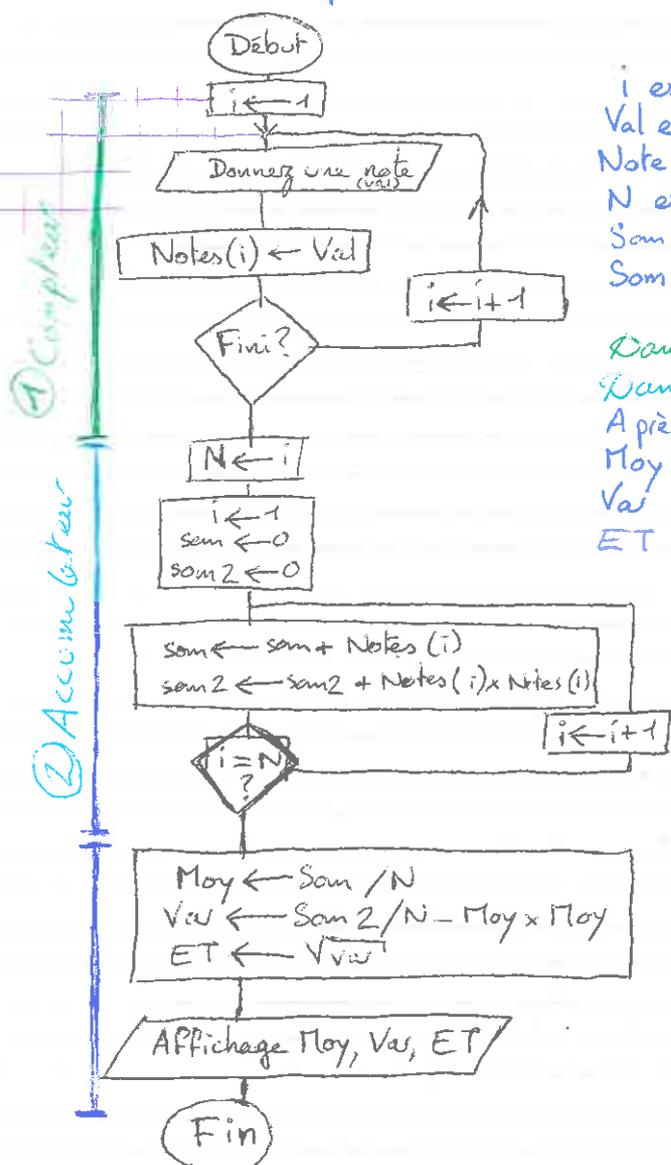
- Dans algorithme "dur"
- Les variables entrées par l'utilisateur sont des variables explicites
- Les variables calculées sont des variables issues de résultats de traitement
- Les variables peuvent être isolées (1 information chacune) $C \leftarrow A + B$
- Les variables peuvent être tableaux Case (1) $\leftarrow 10$ Case (4) $\leftarrow 9,5$

1	2	3	4	5	6	7	8
10	15	2	9,5	9,5	13	5	13

i est l'indice de notes (compteur)
 Val est une variable de stockage temporaire de la note
 Note() est le tableau de stockage
 N est l'effectif total
 Som est l'accumulateur de Notes (variable)
 Som2 est l'accumulateur de Notes²

Dans le compteur : saisie + stockage du tableau
 Dans les accumulateurs : lecture du tableau + sommes
 Après : calcul des paramètres et affichage
 Moy la variable de stockage de la moyenne
 Var la variable de stockage de la variance
 ET la variable de stockage de l'écart type

Ou reprend le compteur avec, impose $i \leftarrow 0$



* Pseudo code

```

DEBUT PROCEDURE
L ← 0
REPETER
    ECRIRE "Donnez une Note"
    LIRE Val
    i ← i + 1
    Notes(i) ← Val
    ECRIRE "Avez-vous terminé? O/N"
    LIRE Reponse
JUSQU A CE QUE Reponse = "O"
N ← i
i ← 0
Som ← 0
Som2 ← 0
    
```

```

REPETER
    i ← i + 1
    Som ← Som + Notes(i)
    Som2 ← Som2 + Notes(i) × Notes(i)
JUSQU A CE QUE i = N
Moy ← Som / N
Var ← Som2 / N - Moy × Moy
ET ← √Var
ECRIRE "la moyenne est:", Moy
ECRIRE "la variance est:", Var
ECRIRE "l'écart-type est:", ET
FIN PROCEDURE
    
```

ISARA-Lyon DSI	TP n°3	Promotion 43
Informatique 2A F. BILLY / O. SAVET	Analyse d'un problème & premiers algorithmes (suite)	Octobre 2011



Exercice 1

Ecrire un algorithme permettant de déterminer si un nombre entré au clavier est (ou non) un nombre premier (nombre divisible uniquement par 1 et par lui-même) :

- réaliser l'organigramme.
- écrire le pseudo-code.

Exercice 2

On souhaite afficher la liste des nombres premiers compris entre 1 et N (N est une valeur limite entrée au clavier) :

- réaliser l'organigramme de cet algorithme.
- écrire le pseudo-code associé.

Exercice 1

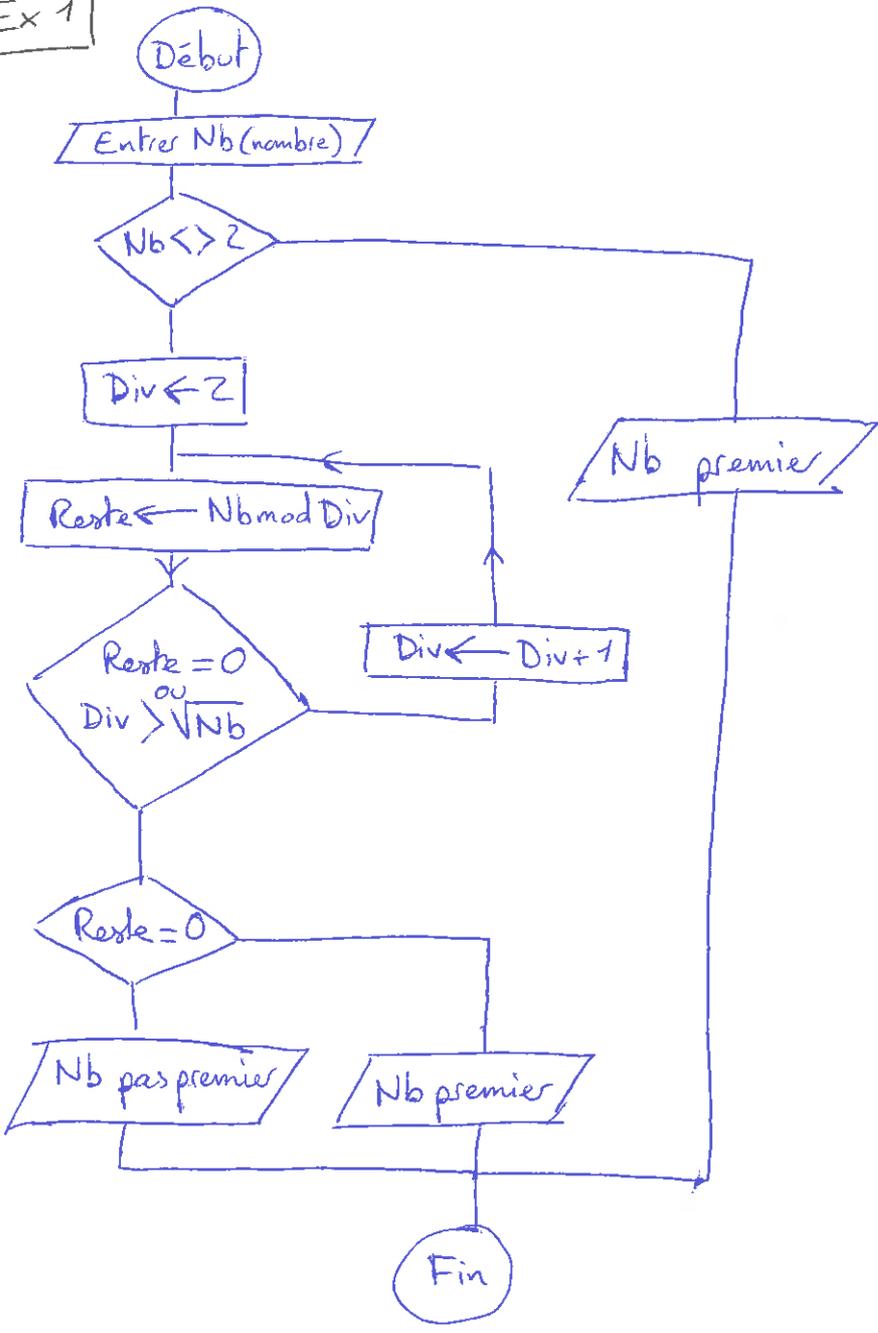
Note : il est inutile de diviser le nombre étudié par un nombre supérieur à la racine carrée de ce nombre étudié.

On étudie...

5 → pas divisible par 2 puis $3^2 = 9$ → premier
 6 → pas div. par 2 → pas premier
 7 → pas div. par 2 puis $3^2 = 9$ → premier
 17 → pas div. par 2; 3; 4 et $5^2 = 25$ → premier
 25 → div. par 5 → pas premier
 41 → pas div. par 2; 3; 4; 5; 6; puis $7^2 = 49$ → premier

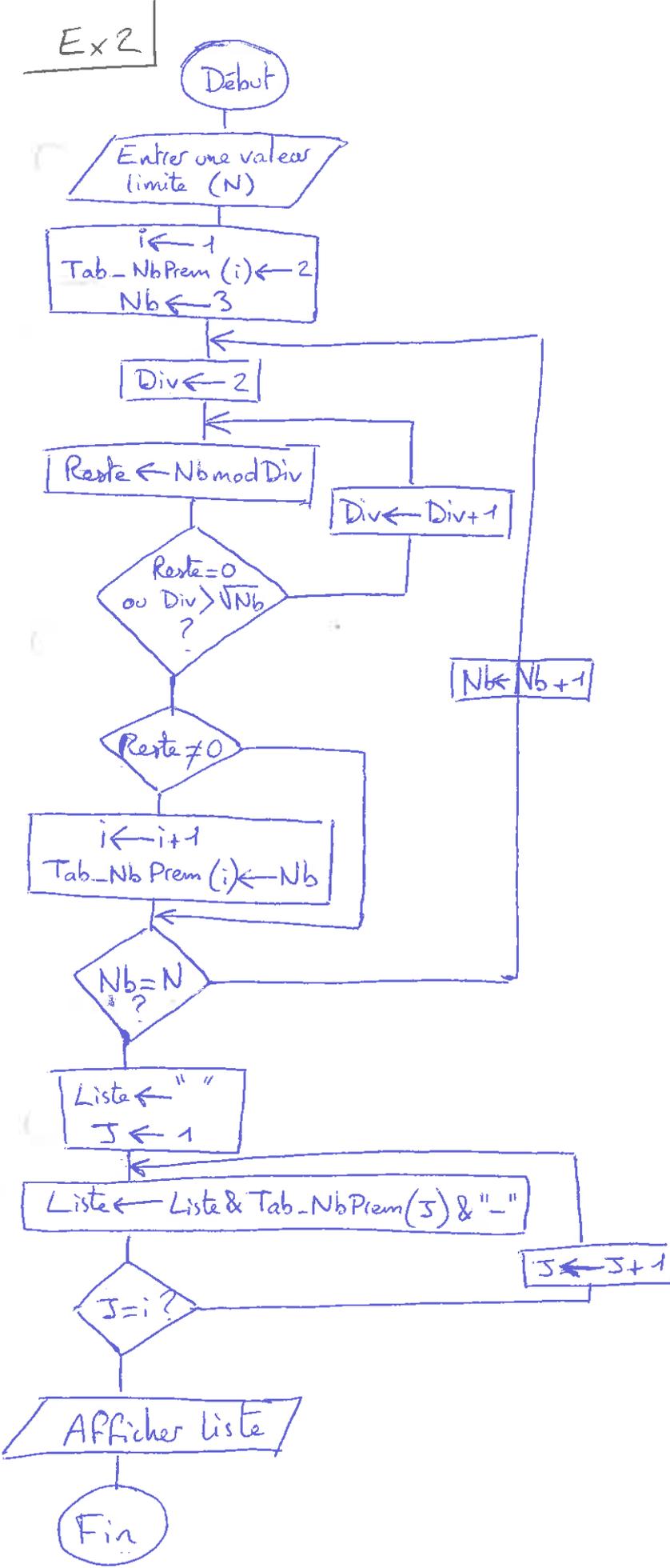
Note = l'exécution de "Nb mod Div" nous donne le reste de la division de "Nb" par "Div".

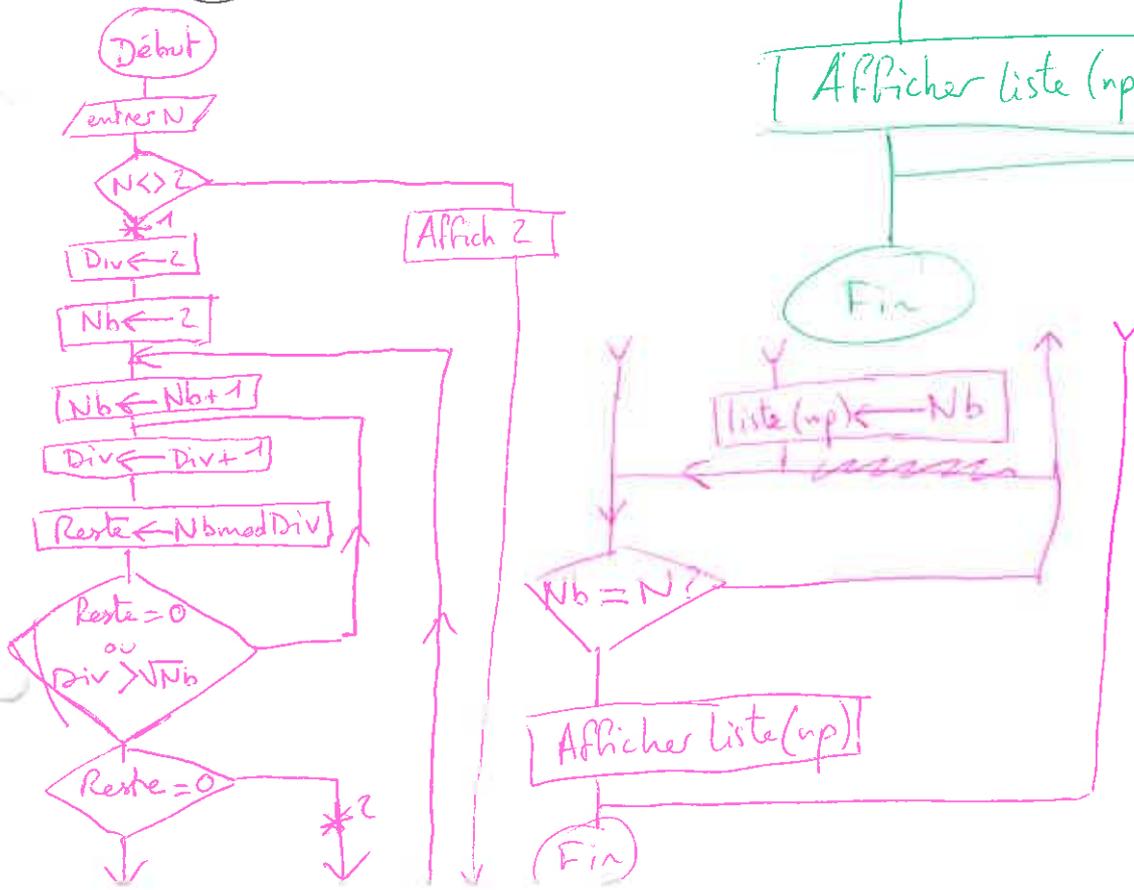
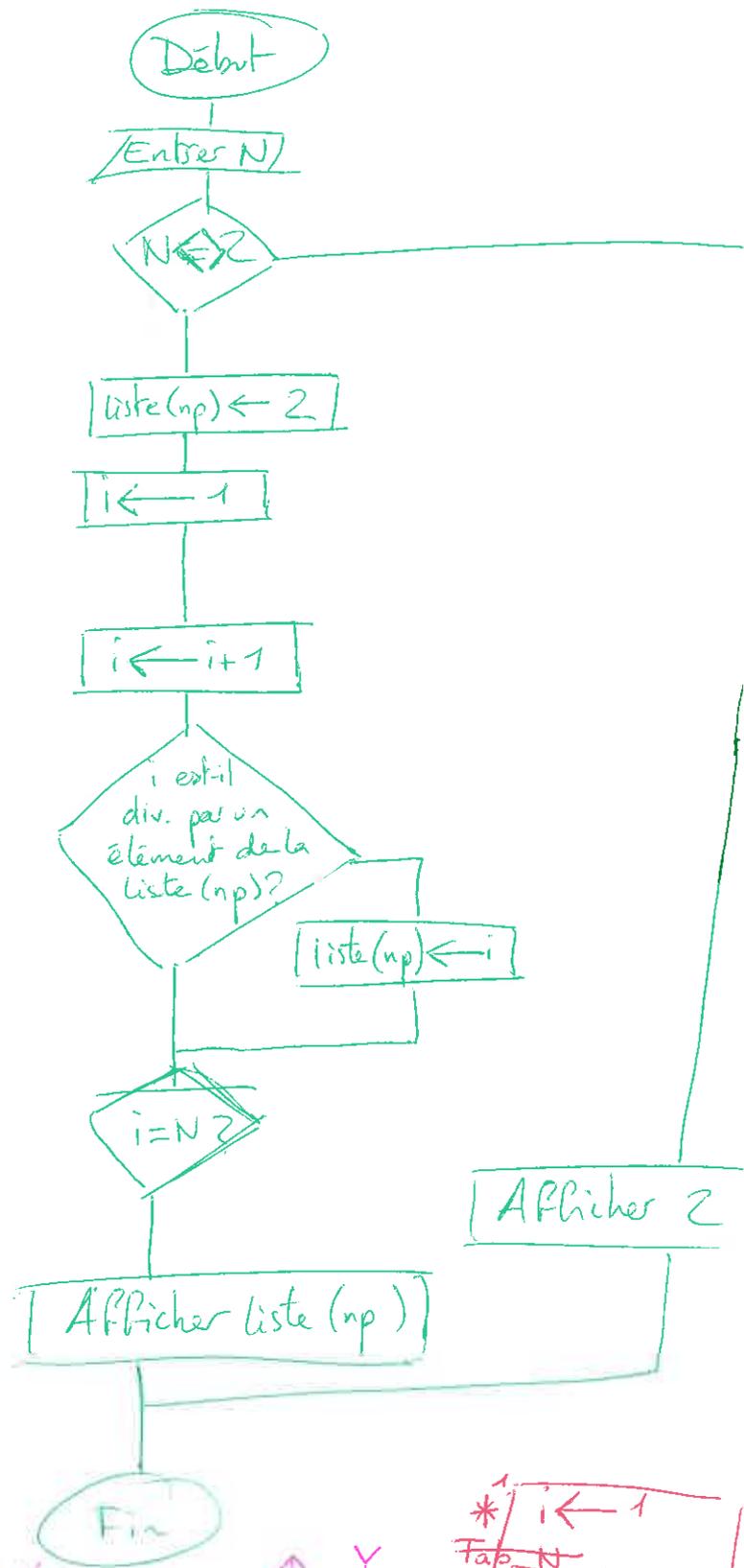
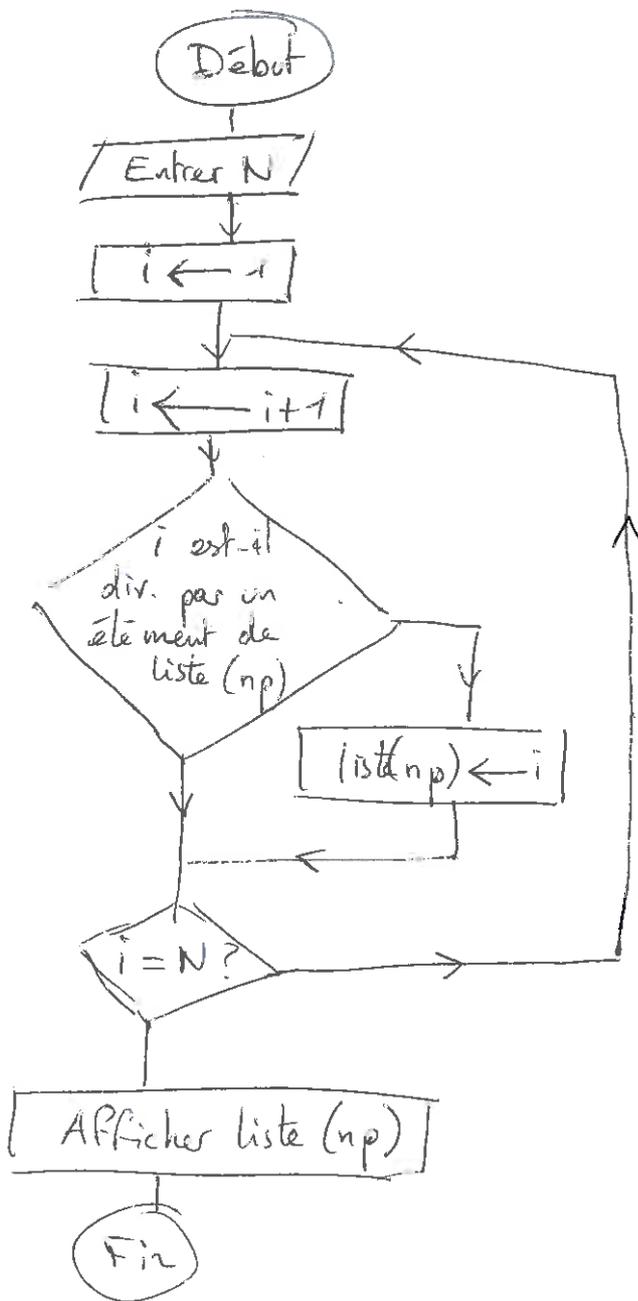
Ex 1



DEBUT PROCEDURE
ECRIRE "Donnez un nombre"
LIRE Nb
SI Nb < 2
ALORS
 Div ← 1
REPETER
 Div ← Div + 1
 Reste ← Nb mod Div
JUSQU'A CE QUE Reste = 0
 ou Div ≥ Nb
Si Reste = 0
ALORS
ECRIRE "Nb pas premier"
SINON
ECRIRE "Nb premier"
FIN SI
SINON
ECRIRE "2 est premier"
FIN SI
FIN PROCEDURE

Ex 2





*1 $i \leftarrow 1$
 $\text{liste}(np)(i) \leftarrow 2$

*2 $i \leftarrow i+1$
 $\text{liste}(np)(i) \leftarrow Nb$



www.lmde.com

LYON
19 RUE DE MARSEILLE

69007 LYON
09.69.369.601 (coût appel local)
N° d'adhérent : 5810303
Assuré : Mle MONSAVOIR ALEXIA
Née le 25.01.1992.
NIR : 292019206427244
Référence : NGUYEN75



MLE MONSAVOIR ALEXIA
15 B VLA PICHON
94600 CHOISY LE ROI

LYON le 21 février 2011

Objet : DOSSIER A COMPLETER

Mademoiselle

Nous sommes au regret de devoir vous retourner votre dossier qui, à ce jour, est incomplet:

DEMANDEZ A VOTRE MEDECIN DE :

- Signer la feuille de soins attestant la prestation de l'acte
- Signer la feuille de soins attestant le paiement de l'acte
- Préciser les Nom/Prénom du patient sur la feuille de soins

AUTRES CAS :

- Faire acquitter la facture :
 - de la pharmacie
 - des frais d'ambulance
 - des frais d'optique
- Faire apposer le cachet de la pharmacie sur la facturation des médicaments
- Faire acquitter la facture des frais de séjour
- AUTRE MOTIF : veuillez adresser la feuille de soins au Centre dont vous dépendiez à la date des soins.....

Nous vous prions d'agréer Mademoiselle, nos salutations distinguées.

LMDE Centre
ACADÉ
Votre Agence

La Mutuelle Des Étudiants - LMDE

• Siège social : 37 rue Marceau - 94200 Ivry-sur-Seine

Pour accéder aux Services d'information proposés par la LMDE, composez le 3260 dites « LMDE » ou le 01 40 92 54 85.
Pour toute demande relative à votre contrat ou au suivi de vos remboursements, composez le 01 40 92 54 85.



ISARA-Lyon DSI	TP n°4 Logique de programmation Découpage d'un problème en sous-problèmes	Promotion 43
Informatique 2A F. BILLY / O. SAVET		Décembre 2011

NOMBRES PARFAITS

Un **nombre parfait** est un entier (naturel) égal à la somme de ses diviseurs (le nombre lui-même étant exclu). A titre d'exemple, 6 est un nombre parfait ($6 = 1+2+3$).

Le but de ce TP est de réaliser un programme permettant d'afficher la liste des nombres parfaits compris entre 1 et une valeur limite L entrée par l'utilisateur.

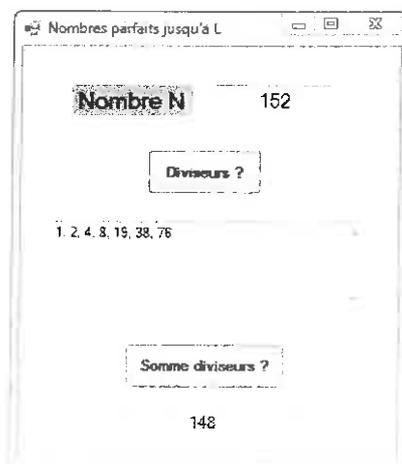
Comme c'est très souvent (pour ne pas dire toujours !) le cas en algorithmique et programmation, nous allons découper la tâche à accomplir en sous-tâches, à savoir :

- a) déterminer quels sont les diviseurs d'un nombre N donné.
- b) faire la somme de ces diviseurs.
- c) tester si cette somme est égale à N, auquel cas N est un nombre parfait.
- d) effectuer a, b et c pour tous les entiers N tels que $1 \leq N \leq L$.

Pour la mise au point du code, vous utiliserez les outils de débogage de la plateforme Visual Studio (points d'arrêts, ajout d'espions pour connaître l'évolution de la valeur des variables, etc...).

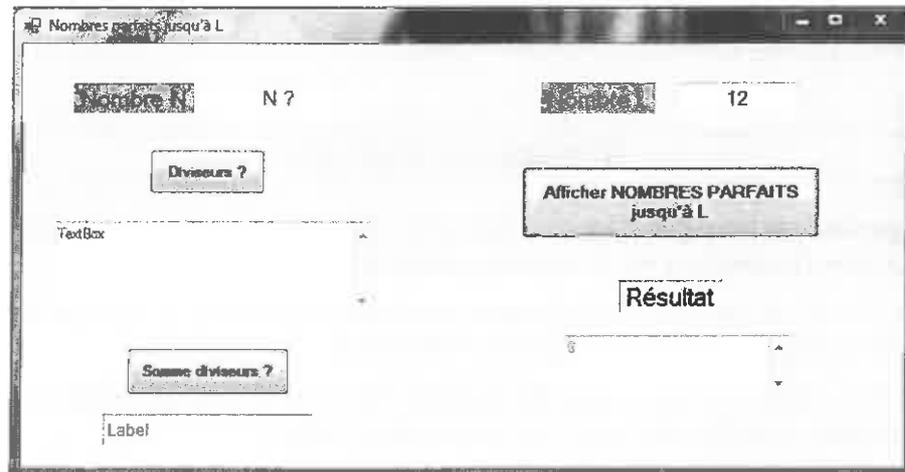
1. Modélisez, sous la forme d'un **organigramme**, un algorithme renvoyant un tableau constitué des diviseurs d'un entier entré par l'utilisateur.
Ecrivez le **pseudo-code** correspondant.
Implémentez le code sous VB.net, à partir de l'IHM qui vous est fournie sur *e-campus* (TP 4 - NbParfaits).

2. Mêmes questions (organigramme et pseudo-code) pour un algorithme permettant de déterminer la somme des éléments d'un tableau, la taille du tableau (c'est-à-dire le nombre d'éléments) étant connue.
Modifiez l'IHM (comme ci-dessous) et implémentez le code.



T.S.V.P. ⇨

3. Modifiez de nouveau l'IHM (comme ci-dessous), puis donnez l'**organigramme** et écrivez le **code** permettant d'obtenir la liste demandée (nombres parfaits compris entre 1 et la valeur limite L).

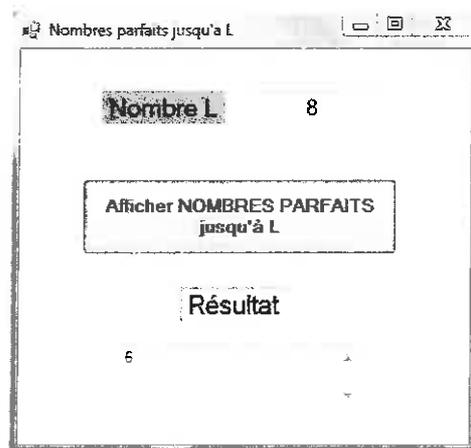


L'organigramme intégrera l'utilisation des algorithmes précédents sous la forme "d'appels" à ceux-ci.

On modélisera ces appels à l'aide du symbole 

De même, pour l'écriture du code, on appellera les fonctions (ou procédures) précédemment écrites en leur passant en argument les variables appropriées.

4. Finalisez votre programme en **redimensionnant** l'IHM de la manière suivante :



Figez ensuite la fenêtre (via les propriétés) pour que le redimensionnement par l'utilisateur soit impossible !

INFO TD4

```

DEBUT PROCEDURE
Ecrire ("Entrez un nombre N")
LIRE N
i ← 0
div ← 1

```

```

REPETER
  div ← div + 1
  reste ← N mod div
  S SI reste = 0
  | tab(i) ← div
  | i ← i + 1
  FIN SI

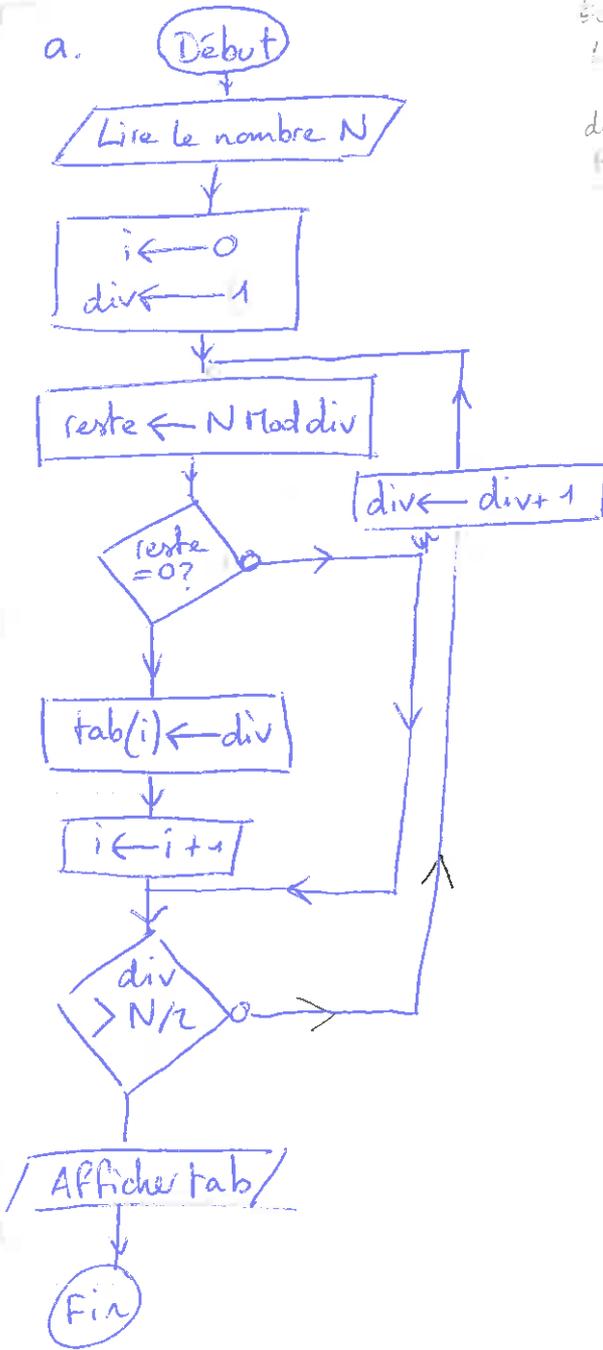
```

JUSQU'À CE QUE $div > N/2$

POUR $j \leftarrow 0$ JUSQU'À $i-1$

 |ste ← list & " " & tab(j)

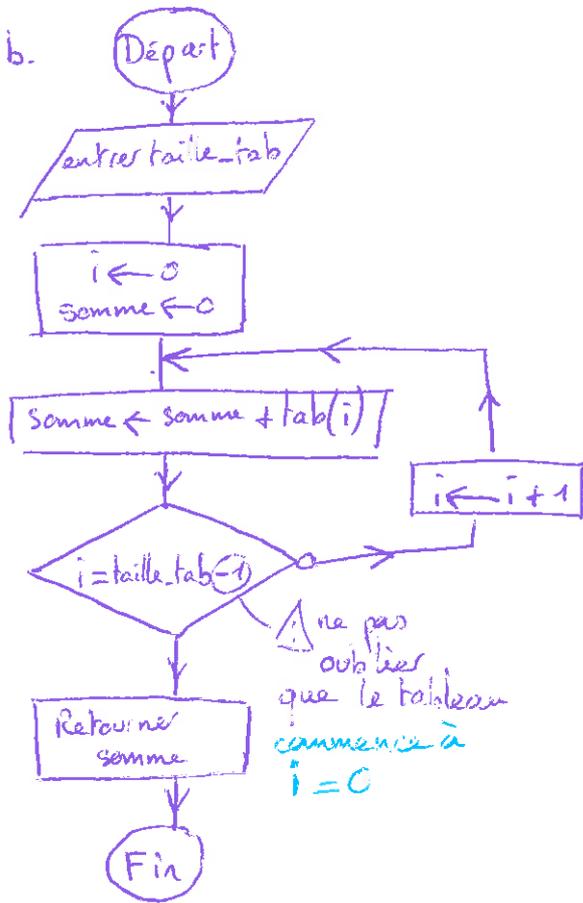
FIN POUR



Types de boucles :

- 1 - jusqu'à ce que
- 2 - tant que
- 3 - pour que

b.



Variable
taille-tab
somme

DEBUT
 LIRE tab, taille-tab
 i ← 0
 somme ← 0
 REPETER
 somme ← somme + tab(i)
 i ← i + 1
 TANT QUE i <> taille-tab - 1
 RETOURNER Somme
 FIN

Faire une fonction sur vb:
 " Public Function somme-tab (tab, taille-tab) As Integer
 Dim somme As Integer

As integer

Return somme

End Function "

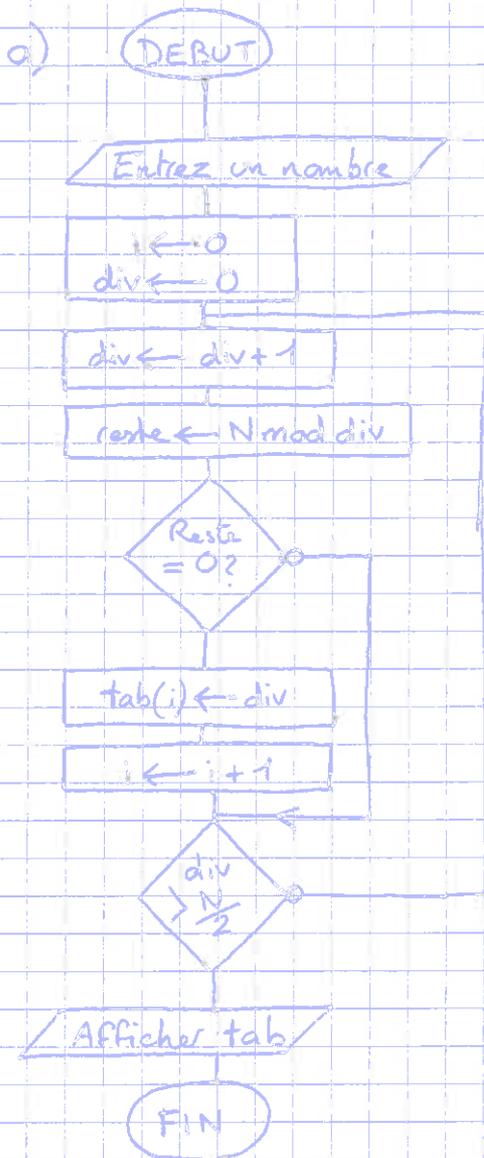
Puis on revient sur l'ancienne Fct.

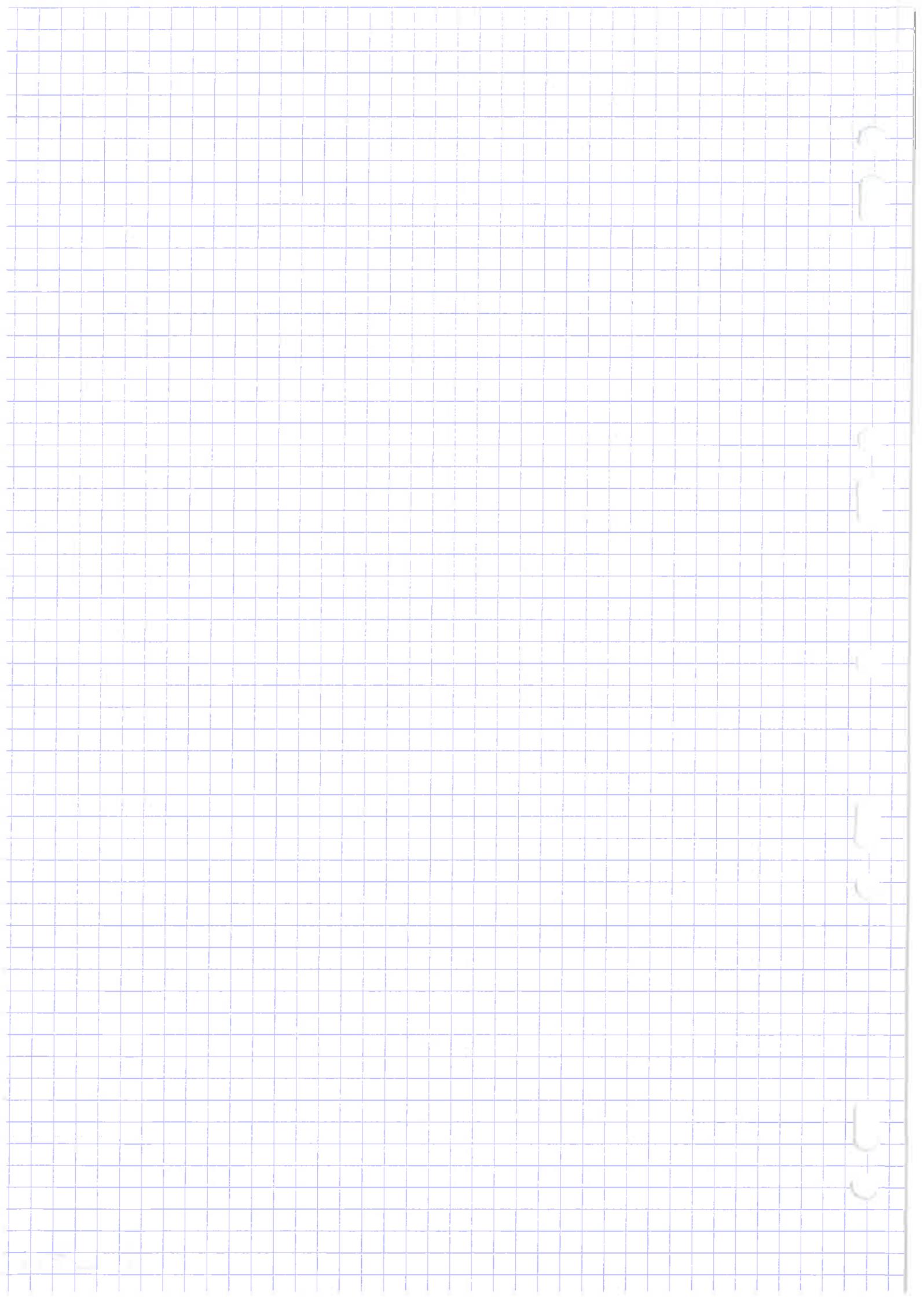
Juste avant l'affiche de la tab, on place la fct somme-tab

"rep = somme.tab (tab.div, tab.div.lenght)

INFORMATIQUE

TD 4





Informatique

Algorithmique

Aucun document ni calculatrice ne sont autorisés pendant l'examen !

Le barème est donné à titre purement indicatif.

La clarté et le soin apporté à la présentation ainsi que la qualité de la rédaction entreront pour une part importante dans l'évaluation de la copie.

Pour pouvoir s'attaquer à l'exercice "bonus", il est impératif d'avoir traité l'exercice 2.

TRES IMPORTANT :

Veillez particulièrement à la lisibilité de vos organigrammes !!!

Avant chaque algorithme, vous préciserez les variables utilisées, leur type (entier, booléen, chaîne de caractères...) et ce à quoi elle servent !

Exercice 1 - Fonction "FaitQuelqueChose" (12 points)

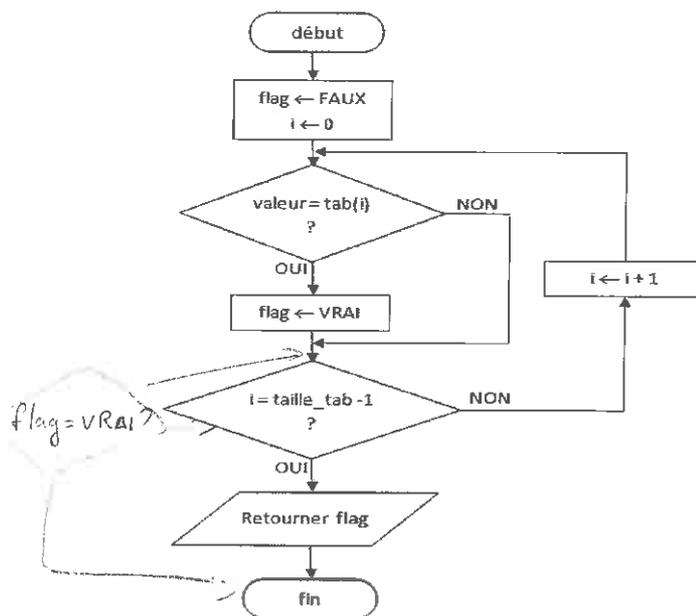
On cherche à coder (en Basic) une fonction dont on connaît :

- la déclaration
- la modélisation sous forme d'un organigramme

La déclaration est la suivante :

```
Function FaitQuelqueChose(ByVal tab, ByVal taille_tab, ByVal valeur)  
    Dim flag As Boolean  
    ' ici on va coder !!!  
    Return flag  
End Function
```

L'organigramme est donné ci-dessous :



T.S.V.P. --->

1. A quoi sert cette fonction ? (soyez concis mais précis)
Trouvez lui un nom plus pertinent que "FaitQuelqueChose".
2. Ecrivez le code de cette fonction en Basic (ou, au choix, utilisez le pseudo-code).
3. Une fonction (dite appelante) utilise la fonction "FaitQuelqueChose" ainsi :

```

Dim tableau(5) As Integer
Dim ret As Boolean

tableau = {-5, 12, 0, 3, 2, -1}
ret = FaitQuelqueChose(tableau, 6, 1)

```

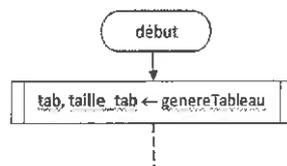
Que vaut "ret" après l'appel de la fonction "FaitQuelqueChose" ? Expliquez...

4. On souhaite maintenant optimiser cet algorithme en sortant de la boucle dès que "flag" a pris la valeur "VRAI".
Refaites un organigramme respectant cette nouvelle contrainte.
5. Ecrivez le code (ou, au choix, utilisez le pseudo-code) associé à votre nouvel organigramme.

Exercice 2 - Permutations (8 points)

Le but de cet exercice est de modéliser un algorithme permettant de **permuter, autant de fois que l'on veut, deux éléments consécutifs d'un tableau** :

- le tableau à traiter (*tab*) est connu à l'entrée de l'algorithme (voir le début d'organigramme ci-dessous) : sa taille (*taille_tab*) et les éléments qui le composent sont fournis par une fonction "genereTableau".
 - l'utilisateur doit saisir l'indice de l'élément qu'il souhaite échanger avec l'élément **suivant**. S'il saisit l'indice du dernier élément du tableau, celui-ci sera échangé avec le premier.
 - l'utilisateur doit pouvoir demander autant de permutations qu'il le souhaite.
 - l'utilisateur doit aussi pouvoir garder le tableau initial tel quel, sans aucune permutation.
 - après chaque permutation, le nouveau tableau sera affiché.
1. Modéliser cet algorithme en complétant l'organigramme suivant.
Pour l'affichage du tableau, vous vous contenterez d'une simple action "afficher tab".



2. Ecrivez le code associé en Basic (ou, au choix, utilisez le pseudo-code).
Cette fois, la boucle d'affichage sera explicite.

Exercice 3 - BONUS

L'algorithme de l'exercice 2 peut être utilisé pour trier un tableau.

Cette manière de trier, par permutation de deux éléments consécutifs, s'appelle un **tri bulle**.

Evidemment, dans le cas d'un tri bulle informatisé, l'utilisateur n'a pas à échanger lui-même les éléments, tout est réalisé par l'algorithme.

A la sortie (de cet algorithme de tri), les éléments sont classés du plus petit au plus grand.

1. Donnez une modélisation, sous forme d'un organigramme, du tri bulle.
2. S'il vous reste encore un peu de temps... codez (en pseudo-code ou Basic) cet algorithme de tri.

ISARA-Lyon DSI	TP n°5	Promotion 43
Informatique 2A F. BILLY / O. SAVET	Initiation aux macros Excel	Mars 2012

I. Création d'un masque avec l'enregistreur de macros

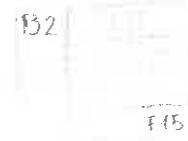
Sous Excel, on souhaite créer un "masque" très simple.

Il s'agit ici d'une zone, dans une feuille de calcul, permettant la **mise en forme automatique des données saisies** dans les cellules de la zone en question.

La zone de travail est composée du groupe de cellules B2 (haut-gauche) à F15 (bas-droite), soit une zone de 14 lignes et 5 colonnes.

La mise en forme à appliquer à cette zone est la suivante :

- fond en gris
- texte rouge, taille 20, pour toutes les cellules
- alignement à droite, avec centrage vertical, pour toutes les cellules
- bordure "double trait" autour de la zone



- 1) Utiliser l'enregistreur de macros pour créer ce masque.
Supprimer la mise en forme de la feuille, puis vérifier que le masque est bien fonctionnel en lançant l'exécution de la macro.
- 2) Ouvrir l'éditeur VBA et observer le code généré. Repérer en particulier les objets utilisés... et tenter de comprendre à quoi ils correspondent !
Si besoin, "nettoyer" ce code, c'est-à-dire supprimer toutes les instructions inutiles.
- 3) Supprimer de nouveau le masque, puis réenregistrer celui-ci (dans une 2^{ème} macro) en effectuant volontairement des manipulations inutiles (en essayant de les mémoriser !).
Repérer dans le code VBA les instructions sans intérêt pour la création du masque, puis effectuer le "nettoyage".

II. Amélioration du masque – programmation VBA

On se propose d'ajouter des contrôles sur la feuille de calcul et d'utiliser l'éditeur VBA pour associer des procédures (donc "du code") à ces contrôles.

Pour commencer, nous allons ajouter l'onglet "Développeur" à l'environnement du tableur. Faire apparaître cet onglet (Options Excel → Standard).

- 1) Créer un bouton (avec un libellé explicite !), sur la feuille de calcul, permettant d'activer le masque précédent.
- 2) Créer un autre bouton permettant de supprimer ce masque.
- 3) Créer un 3^{ème} bouton pour vérifier que toutes les données entrées dans les cellules de la zone (correspondant au masque) sont bien **de type numérique**, et **comprises entre 0 et 100**.

Informez l'utilisateur (en utilisant une *MsgBox*) du nombre de données "illicites" !

NB : pour cette question, le code sera **entièrement écrit** en utilisant l'éditeur VBA !!

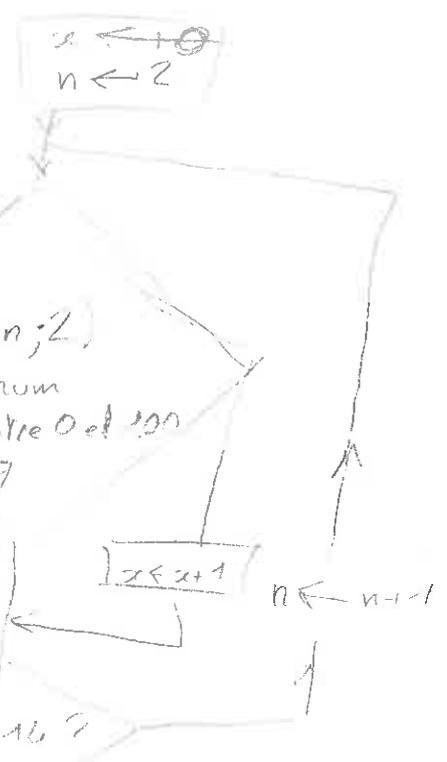
x ← 'est les illicites
 n variable
 (ligne)
 IF



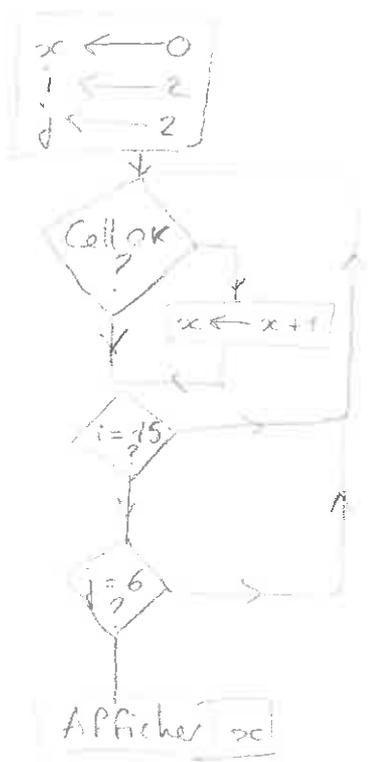
DE a f 15
 → a 1000
 → 0 à 100

IF Cells(n;2) =
 num
 AND Cells(n;2) > 0
 AND Cells(n;2) < 100
 !ELSE
 $x \leftarrow x + 1$

Cells(n;2)
 = num
 entre 0 et 100



$n = 16$
 Cells(n;2)
 = 16



ISARA-Lyon DSI	TP n°6	Promotion 43
Informatique 2A F. BILLY / O. SAVET	Macros Excel	Mars 2012

I. Mise à jour d'un tableau pluriannuel

La société "Mov'Info" souhaite réaliser un tableau pluriannuel (**sur les 4 dernières années**) de ses dépenses. Ce tableau est mis à jour chaque année. Lorsque les dépenses d'une nouvelle année sont connues, l'utilisateur du tableau doit :

- effacer le contenu de la colonne des dépenses de la première année.
- copier le contenu des trois dernières colonnes à partir de la première.
- saisir les dépenses de la nouvelle année.

On se propose de développer un programme qui automatisera, chaque année, la mise à jour de ce tableau.

- 1) Saisir sous Excel le tableau ci-dessous, **en respectant sa mise en forme**.

	A	B	C	D	E
1					
2	ETAT DES DEPENSES				
3	POSTES	2007	2008	2009	2010
4	Achat de matériel	125000	85000	98000	241510
5	Autres achats	32100	21540	158000	96000
6	Main d'œuvre	5800	16000	321557	15847
7	Charges sociales	69000	65400	25500	142100
8	Intérêts emprunts	5000	3300	4900	2500
9	TOTAL DES DEPENSES	238900	191240	687957	487957
10					

- 2) Faire apparaître l'onglet "Développeur" (Options Excel → Standard).

A l'aide de l'enregistreur, créer une macro "InsérerColonne" (avec raccourci *Ctrl+Maj+N*) permettant d'automatiser les manipulations pour la saisie des données 2011 :

- supprimer les données 2007 (sans supprimer la colonne "B").
- utiliser le "Collage spécial" afin de recopier **uniquement les valeurs** des données 2008 à 2010. Evidemment, on ne touchera pas à la ligne "TOTAL DES DEPENSES" afin de conserver le calcul de ce total.
- supprimer les données de la colonne 2010 en doublon.
- sur la ligne 3, la nouvelle année sera calculée automatiquement...

Le tableau obtenu sera ainsi le suivant :

	A	B	C	D	E
1					
2	ETAT DES DEPENSES				
3	POSTES	2008	2009	2010	2011
4	Achat de matériel	85000	98000	241510	
5	Autres achats	21540	158000	96000	
6	Main d'œuvre	16000	321557	15847	
7	Charges sociales	65400	25500	142100	
8	Intérêts emprunts	3300	4900	2500	
9	TOTAL DES DEPENSES	191240	687957	487957	
10					

→ 3) Saisir les données 2011, par exemple : 150200, 90000, 140000, 55000, 10000.

→ 4) Documenter **chaque ligne** de la macro, c'est-à-dire ajouter un commentaire précisant l'intérêt de celle-ci. On en profitera, si besoin, pour "nettoyer" le code généré.

→ 5) Ajouter, sur la feuille de calcul, un bouton "Saisie de l'année en cours", puis associer le code à ce bouton.

6) Ajouter un deuxième bouton enclenchant le **calcul des dépenses sur les 4 dernières années** : le résultat mis en forme (avec un intitulé explicite) apparaîtra sur la feuille de calcul (où vous le souhaitez !).

7) Ajouter un troisième bouton faisant disparaître ce qui a été créé par le bouton précédent.

T.S.V.P. →

II. Création d'un formulaire de saisie

Il s'agit ici de remplir un tableau à partir d'un **formulaire** (fenêtre de saisie, comparable aux IHM créées sous Visual Studio).

- 1) Récupérer sur e-campus le classeur "*formulaire_saisie.xlsx*".
Ce classeur inclut une feuille "*Annuaire*" (c'est elle qu'on va remplir via le formulaire) et une feuille "*Pays*" à laquelle on va se référer pour mettre à jour le formulaire.
- 2) Ouvrir l'éditeur VBA, puis créer le formulaire suivant :



Attention : pour fonctionner, les "**boutons radio**" (en anglais : "radio buttons" ou "option buttons") doivent être intégrés à un groupe, c'est-à-dire inclus dans un cadre (en anglais : "frame").

- 3) Sur la feuille "*Annuaire*", créer un bouton "Ajouter un contact", puis associer à celui-ci une macro permettant de faire apparaître le formulaire (méthode *Show* à appliquer à l'objet).
- 4) Le bouton "Fermer" doit permettre de faire disparaître le formulaire, mais aussi de vider le contenu des contrôles.
Dans la macro associée, appliquer la méthode *Hide* au formulaire. Que constatez-vous lorsque vous refaites "Ajouter un contact" ?
(Tester *Unload me* dans cette même macro.)
- 5) La **liste déroulante** ("**combo box**") de choix du pays doit être remplie à l'ouverture du formulaire :
 - a. créer une macro associée à l'événement *Initialize* du formulaire.
 - b. dans cette macro, effectuer une boucle de lecture de toutes les données de la première colonne de la feuille "*Pays*". On utilisera la méthode *AddItem* de la liste déroulante.
- 6) Définir l'**ordre de tabulation** des contrôles du formulaire afin d'en faciliter la saisie.
- 7) Créer une macro associée à l'événement *Click* du bouton "Ajouter". Dans cette macro :
 - a. insérer les valeurs des champs "Nom", "Prénom", "Adresse", "Lieu" et "Pays" dans la feuille "*Annuaire*", sur la 1^{ère} ligne.
 - b. **boucler** sur les **boutons radio** (utiliser "for each... in") pour récupérer l'information "civilité", puis stocker celle-ci dans la feuille "*Annuaire*" (sur la 1^{ère} ligne également).
- 8) Pour que ce formulaire de saisie soit intéressant, il faut évidemment, à chaque nouveau contact, que celui-ci soit reporté **sur la 1^{ère} ligne vide** de la feuille "*Annuaire*".
Créer dans la procédure précédente une variable "*no_ligne*", puis imaginer une méthode pour mettre à jour celle-ci à chaque ajout d'un contact.
- 9) Pour que cet exercice soit complet, on se propose maintenant de vérifier, à chaque ajout de contact, si tous les champs du formulaire sont effectivement remplis. Si ce n'est pas le cas, **on basculera en rouge (et en gras !)** les **labels** associés au(x) champ(s) vide(s), et on ne stockera évidemment pas les données.
Compléter la procédure précédente afin d'intégrer cette fonctionnalité ; on utilisera la propriété *ForeColor* des objets et la fonction RGB(r,g,b).

ISARA-Lyon DSI	TP n°7 Initiation SGBD sous Access : tables, champs, enregistrements, formulaires, requêtes simples	Promotion 43
Informatique 2A F. BILLY / O. SAVET		Avril 2012

I. Création d'une base de données

- Lancer le logiciel "Microsoft Access", puis créer une base de données vide : celle-ci sera nommée "CDthèque.accdb".
- Par défaut, une table "Table1" est automatiquement créée.
Renommer cette table (→ "Albums"), puis créer dans celle-ci les 5 champs suivants (en mode "Création") :
 - **Code classement** (type "texte", 10 caractères maxi)
 - **Titre album** (type "texte", 255 caractères maxi)
 - **Année album** (type "numérique", entier)
 - **Interprète** (type "texte", 255 caractères maxi)
 - **Titre1** (type "texte", 255 caractères maxi)
- NB** : le champ "**code classement**" sera composé de 3 lettres représentant la catégorie (POP pour la musique pop, FOL → folk, ROC → rock, etc...), d'un tiret, puis d'un nombre à 6 chiffres. Par exemple, "POP-001269" est le 1269^{ème} album de la catégorie "POP".
- Enregistrer la base "CDthèque" (**attention** : n'utilisez pas le chemin proposé par défaut, stockez votre fichier dans votre dossier réseau personnel → \\netappix\perso\$\promo43\...).
- Basculer en mode "**Feuilles de données**" et entrer **trois enregistrements** (idéalement de catégories différentes) correspondant à vos goûts musicaux (évidemment, vous pouvez consulter le web !).
- Ajouter ensuite à cette table "**Albums**" les trois enregistrements suivants (en respectant la casse) :
 - FOL-000121 / Slow Train Coming / 1979 / Dylan, Bob / Gotta Serve Somebody
 - ROC-000001 / Sleeping with Ghosts / 2003 / Placebo / Bulletproof Cupid
 - PUN-000001 / Sandinasta / 1977 / The Clash / The Magnificent Seven
- La table "**Albums**" contient maintenant **6 enregistrements**.
- Enregistrer la base modifiée, puis en créer une copie de sauvegarde.

II. "Fusion" de deux tables

On souhaite à présent récupérer des données stockées dans une autre base nommée "CDthèque_bis". Celle-ci inclut une table "CDs" comportant 5 champs identiques à ceux que vous avez créés dans la table "Albums".

Le but est ici de regrouper tous les enregistrements dans une seule et même table. Nous allons réaliser ceci en 2 étapes : **importation** de la table "CDs" dans la base "CDthèque", puis **fusion** des tables "Albums" et "CDs".

- **Copier dans votre dossier réseau personnel** la base "CDthèque_bis", disponible sur e-campus (S4_INFO \Documents et Liens \ TP \ TP 7).
- Ouvrir la base "CDthèque", puis utiliser l'assistant d'importation (onglet *Données externes* → *Access*) : la table "CDs" apparaît ainsi dans l'explorateur d'objets.
- Faire une copie de la table "Albums" (clic droit sur la table, dans l'explorateur d'objets → table "Copie de Albums").

La fusion des tables va être réalisée à l'aide d'une **requête action** de type "**ajout**" (nous reviendrons sur la notion de requête plus tard dans ce TD). Nous allons ainsi compléter la table "Albums" avec les enregistrements de la table "CDs".

- Cliquer sur l'onglet *Créer*, puis activer la commande *Création de requête*.
- Afficher la table "CDs" (table à copier !), puis sélectionner le type :
- Pour récupérer **toutes** les informations de la table, indiquer **CD.*** sur la ligne concernant le champ à traiter (* signifie que tous les champs sont pris en compte), puis spécifier la table destination. La grille de définition de la requête se présente donc comme suit :

```

Champ : CDs
Table : CDs
Tri :
Ajouter à : Albums
Critères :
Ouvr

```

- Enregistrer la requête (on pourra par exemple l'appeler "Fusion"), puis lancer son exécution
- Analyser les problèmes rencontrés lors de la fusion, puis effectuer les modifications nécessaires de manière à avoir tous les enregistrements attendus dans la table "Albums".
- Quel autre problème subsiste-t-il ? Y remédier, puis supprimer la table "CDs"

III. Création d'un formulaire simple

Un Système de Gestion de Bases de Données (SGBD) intègre des outils qui permettent de créer des interfaces graphiques conviviales pour l'entrée de nouveaux enregistrements dans la base.

Ces interfaces sont appelées **formulaires**.

- Créer un formulaire convivial (vous pouvez même ajouter quelques images si vous le souhaitez...) pour la saisie des informations concernant les nouveaux albums à entrer dans la base.
- Tester le formulaire créé en ajoutant quelques enregistrements.

Saisie de nouveaux albums

RAN 00001
 Led Zepplin IV
 1971
 Led Zepplin
 Black dog

IV. Quelques requêtes simples...

Les SGBD comprennent tous un outil primordial et extrêmement puissant : la **requête**.

Celles-ci vont servir, par exemple, à afficher **uniquement certaines données** contenues dans les tables **selon certains critères**.

Elles peuvent aussi faire des calculs sur vos données, ainsi que créer des sources de données pour les formulaires ou même d'autres requêtes (on peut faire une requête sur le résultat d'une autre requête !).

Elles servent encore à modifier des tables existantes ou éventuellement à en créer des nouvelles.

On se propose donc maintenant de tester quelques requêtes très simples sur notre table "Albums" :

- trier les albums par date.
- trier les albums par interprète.
- extraire les albums sortis avant 1980.
- extraire les albums de rock (code ROC-*).
- extraire les albums de rock sortis avant 1980 :
 - à partir de la table "Albums"
 - à partir des résultats de la requête précédente

Initiation à l'BI sous Access

On va étudier les bases de données, notamment avec MERISE.

Y a des concepts de tables et relations.

ex : On appelle une table Album, dans laquelle on a des champs

NB: une fois la table créée, elle reste qd m vide.

ALBUMS	
code classement	
titre album	
année album	
interprète	
titre 1	

claf
primaire

text
text
entier
text
text

Dim "titre" ≈ "objet"
intég ≈ "classe"

Ainsi, un enregistrement de la table album peut alors être

MET-000000 | In Waves | 2011 | Trivium | Capzing The Sea

↑
type de musique

SOUS MICROSOFT ACCESS

(cf. les consignes)

Informatique

Programmation VBA sous Excel

Ce travail est à réaliser **par binôme**.

Vous inclurez **obligatoirement** des commentaires dans votre code, et indenterez celui-ci.

ATTENTION : lors de vos recherches sur le "web", vous trouverez certains éléments qui vous aideront à réaliser ce travail. Comme pour la maquette réalisée au S3, il ne s'agit pas simplement de copier/coller du code, mais bien de vous **approprier** ces méthodes, c'est-à-dire de les comprendre, d'en retenir les grands principes et de les adapter au sujet ci-dessous.

Tout plagiat manifeste du travail d'un autre binôme sera durement sanctionné !

Réalisation d'un menu personnalisé pour le calcul de paramètres statistiques

Ce travail consiste, à partir d'une feuille de calcul Excel prédéfinie, à écrire des macros permettant le calcul, via un menu personnalisé, de différents paramètres statistiques.

Fichier de départ et fichier final

Vous travaillerez à partir du fichier nommé "Maquette Excel VBA - sujet (contingence).xlsm" accessible sur e-campus (INFO_S4).

ATTENTION : enregistrez-le sur le réseau (dans votre espace personnel) ou sur une clef USB; ne travaillez pas directement sur le fichier d'e-campus! (autrement dit, **ne pas faire "Ouvrir avec..."**).

TRES IMPORTANT : votre fichier final (avec ajout des macros que vous aurez écrites!) sera nommé "Contingence_Nom1_Nom2.xlsm", en remplaçant évidemment "Nom1" et "Nom2" par vos noms respectifs.

Maquette "exemple"

Sur e-campus (INFO_S4), vous avez à votre disposition le fichier "Maquette Excel VBA (corr - PROTECTED).xlsm" qui vous donne une idée de ce que doit être votre fichier final.

Cette maquette n'est pas un modèle à suivre absolument, vous pouvez donc vous en "écarter", du moment que les différents paramètres imposés sont effectivement calculés. Notez bien que d'autres paramètres statistiques peuvent être traités par votre maquette (si vous avez le temps!).

Evidemment, le code de cette maquette est protégé, donc non accessible!

SUJET :

- A l'ouverture du classeur, un onglet "Compléments" apparaît dans la barre de menus; le menu personnalisé "Outils pour statistiques" y est automatiquement chargé. ✕

Pour information, le code associé à ce chargement automatique est situé (sous VBE) dans "ThisWorkbook" (dossier "Microsoft Excel Objects"). **NE PAS MODIFIER CE CODE!!!**

- Codez d'abord les 3 procédures du module "ProcéduresDuMenu".

- **Le contenu du module "BarreStat" ne doit pas être modifié!**

- Ecrire ensuite "en parallèle" le code associé au formulaire et les procédures (ou fonctions) du module "Outils". Une procédure de ce module "Outils", nommée par exemple "**Fonc**", peut être appelée à partir du code du formulaire par "**Outils.Fonc**".

- N'hésitez pas à compléter ce mini-projet si vous avez des idées **pertinentes!**

* avec 3 boutons

T.S.V.P. -->

TRAVAIL A RENDRE :

- un "mini-dossier" (4 ou 5 pages maximum) comprenant un **manuel d'utilisation** succinct de votre programme (pensez à l'illustrer avec quelques copies d'écran). Vous préciserez également quelles sont les limites de votre application, et les idées que vous pourriez avoir pour l'améliorer (même si ces améliorations vous paraissent très difficiles à mettre en oeuvre...).
- le fichier "Contingence_Nom1_Nom2.xlsm".

Le "mini-dossier" est à rendre à la scolarité au plus tard le mercredi 23/05/2012 avant 12h00.

Le fichier Excel est à déposer sur e-campus (INFO_S4, espace "Travaux") au plus tard le mercredi 23/05/2012 avant 18h00.

*Dossier → organigramme des fonctions intéressantes
→ limites du projet
→ autres idées (développées ou non)*

Informatique

Programmation

*La clarté et le soin apporté à la présentation entreront pour une part importante dans l'évaluation.
Pour la partie "programmation", le code sera **OBLIGATOIREMENT** commenté et correctement indenté !!*

ATTENTION : si, pour certains algorithmes, vous vous inspirez d'idées trouvées sur le "web", ne perdez pas de vue qu'il faut nécessairement vous approprier ces algorithmes, c'est-à-dire les comprendre et les adapter aux fonctions et procédures qui vont leur être imposées ainsi qu'à votre IHM.

Tout plagiat manifeste sera durement sanctionné !

Réalisation d'une maquette logicielle de cryptage

Ce travail consiste à **créer un programme** capable de **crypter et décrypter** un message de type "texte".
Pour le cryptage, le message à traiter sera entré par l'utilisateur dans la fenêtre appropriée, puis éventuellement enregistré sur le disque dur (fichier .txt).

Pour le décryptage, le message à traiter sera :

- soit entré par l'utilisateur sur l'interface graphique.
- soit extrait d'un fichier texte (.txt) enregistré sur le disque dur.

Dans le cas du stockage, le fichier à crypter ou à décrypter ne se trouvera pas à un endroit précis sur le disque : il faut donc pouvoir parcourir l'arborescence de celui-ci.

Programme fourni

Vous trouverez sur e-campus un programme complet (sans le code évidemment) de la maquette finalisée.
L'ergonomie de votre IHM devra fortement s'en inspirer (apparition de boutons, désactivation de fenêtres, etc...).

Vous n'êtes néanmoins pas tenu d'avoir exactement la même interface : si vous avez des idées pertinentes, n'hésitez pas ! Veillez cependant à ce que votre IHM soit sobre, complète et répondant au mieux aux objectifs à réaliser par le programme.

Organisation des données

Les données seront essentiellement stockées dans des variables "texte" (String) et dans des tableaux d'entiers.
Evidemment, vous pouvez utiliser autant de variables que vous le souhaitez.

Un peu de théorie...

Le cryptage/décryptage est basé, pour chaque caractère du message "clair" (donc à crypter) ou du message crypté, sur la récupération de son **code ASCII**. Les messages clairs et les messages cryptés ne seront composés que de caractères dont ce code est tel que $32 \leq \text{code} \leq 126$, donc du caractère "espace" (codé 32) au "tilde" (codé 126).

95 caractères sont donc potentiellement utilisables ; la liste des codes ASCII se trouve facilement sur Internet (www.table-ascii.com).

Principe du cryptage

- A chaque caractère, on va associer un entier **entre 0 et 94**, en suivant l'ordre du codage ASCII.
On aura ainsi une correspondance caractère/entier comme suit :

code ASCII



-32

Esp.	!	"	#	...	0	1	2	...	A	B	C	...	a	b	c	...		}	~
0	1	2	3	...	16	17	18	...	33	34	35	...	65	66	67	...	92	93	94

• A partir de cette correspondance, on va coder le texte clair. Par exemple, si celui-ci est "Moi, je fais de l'info!" :

texte	M	o	i	,		j	e		f	a	i	s		d	e		l	'	i	n	f	o		!
codage	45	79	73	12	0	74	69	0	70	65	73	83	0	68	69	0	76	7	73	78	70	79	0	1

• L'astuce, pour rendre le texte quasi-indéchiffrable, consiste maintenant à utiliser ce qu'on appelle une **clef de cryptage**. Celle-ci peut être une phrase, voire même un simple mot.

A partir de cette clef, on va créer un deuxième tableau de même taille que le texte à crypter constitué de la clef de cryptage répétée autant de fois que nécessaire! Par exemple, si la clef est "2A UE_INFO" :

Mot "clef"	2	A		U	E		I	N	F	O	2	A		U	E		I	N	F	O	2	A		U
codage	18	33	0	53	37	63	41	46	38	47	18	33	0	53	37	63	41	46	38	47	18	33	0	53

• On va ensuite créer un autre tableau dont chaque élément sera la somme des codes des éléments correspondants dans les 2 tableaux précédents (ce dernier tableau sera donc encore de même taille!). Lorsqu'une somme dépasse 94, on "revient à 0" : on applique donc cette somme modulo 95.

Une fois ce nouveau tableau constitué, on convertit ces codes en caractères avec le tableau de correspondance présenté précédemment.

Ainsi, pour notre exemple :

Somme	63	112	73	65	37	137	110	46	108	112	91	116	0	121	106	63	117	53	111	125	88	112	0	54
codage	63	17	73	65	37	42	15	46	13	17	91	21	0	26	11	63	22	53	16	30	88	17	0	54
Caractère correspo.	_	1	i	a	E	J	/	N	-	1	(5		:	+	_	6	U	0	>	x	1		V

Le message crypté sera donc "_liaEJ/N-15 :+_6U0>x1 V"

• En analysant la méthode de cryptage, à vous de trouver celle utilisée pour le décryptage!

Consignes particulières

Les caractères accentués (à, é, è, ...) ne seront jamais utilisés (ni dans les textes à crypter, ni dans ceux à décrypter) : c'est logique puisque leurs codes ASCII est supérieur à 126.

De même, un texte à crypter ou à décrypter ne devra pas comporter un appui sur la touche "Entrée" (CR, code ASCII : 13).

La phrase à crypter ou à décrypter n'excèdera pas 300 caractères (police "Lucida Console" conseillée car la largeur est fixe pour tous les caractères).

La clef de cryptage n'excèdera pas 50 caractères.

L'interface graphique (IHM) aura une taille figée et non "maximisable". En revanche, elle pourra être réduite dans la barre des tâches.

Evolutivité

De même que pour l'IHM, si vous avez des idées de fonctionnalités à ajouter à cette maquette, n'hésitez pas. Mais, encore une fois, ne perdez pas de vue qu'il s'agit D'ABORD de coder les fonctionnalités demandées... AVANT d'en ajouter d'autres!!!

TRAVAIL A RENDRE :

- un "mini-dossier" (5 ou 6 pages maximum) comprenant le(s) organigramme(s) et le(s) pseudo-code(s) que vous jugerez importants, ainsi qu'un **manuel d'utilisation** succinct de votre programme. Vous préciserez également quelles sont les **limites** de votre application et donnerez toutes les informations utiles à la bonne compréhension de vos choix.
- le projet développé en Basic sous Visual Studio.

Le "mini-dossier" est à rendre à la scolarité au plus tard le vendredi 10/02/2012 avant 12h00.

Le programme (exécutable) ET le code (TOUS les fichiers) sont à déposer sur e-campus (espace "Travaux") au plus tard le vendredi 10/02/2012 avant 16h00.

All along the walk, the marmot saw things she had never sown before, she was so happy.

Suddenly, the sow stopped, petrified: a few meters further, men were loud speaking. The marmot knew men were dangerous, but this day, she wanted to take a closer look. She approached them, as discreetly as she could.

But they saw her.

So they began to speak very quietly, to squat down and whistle.

One of them reached out to give her something to eat. She smelt it, tasted it, liked it, and ate it. Then, she thought humans were so a strange specie, and decided to come back quickly. She jumped inside the saucepan, and the sow run away.

At the end of the day, the sow took the marmot back to her hole. The marmot took down the saucepan, thanked the sow, and said goodbye.

We can't finish this story within tell you what happened with the two men. They were a quad bike's amateur who wanted to convert the hill into a circuit, and a friend who wanted him to understand that there were protected species on this place.

And, seeing the marmot, the quad bike's amateur finally accepted to search another place where to create his circuit within damaging to much the nature.

And that's how, waking up earlier, the marmot didn't saw her life turning into her nightmare. As the early bird catches the worm... the early marmot catches the sow with a saucepan.

