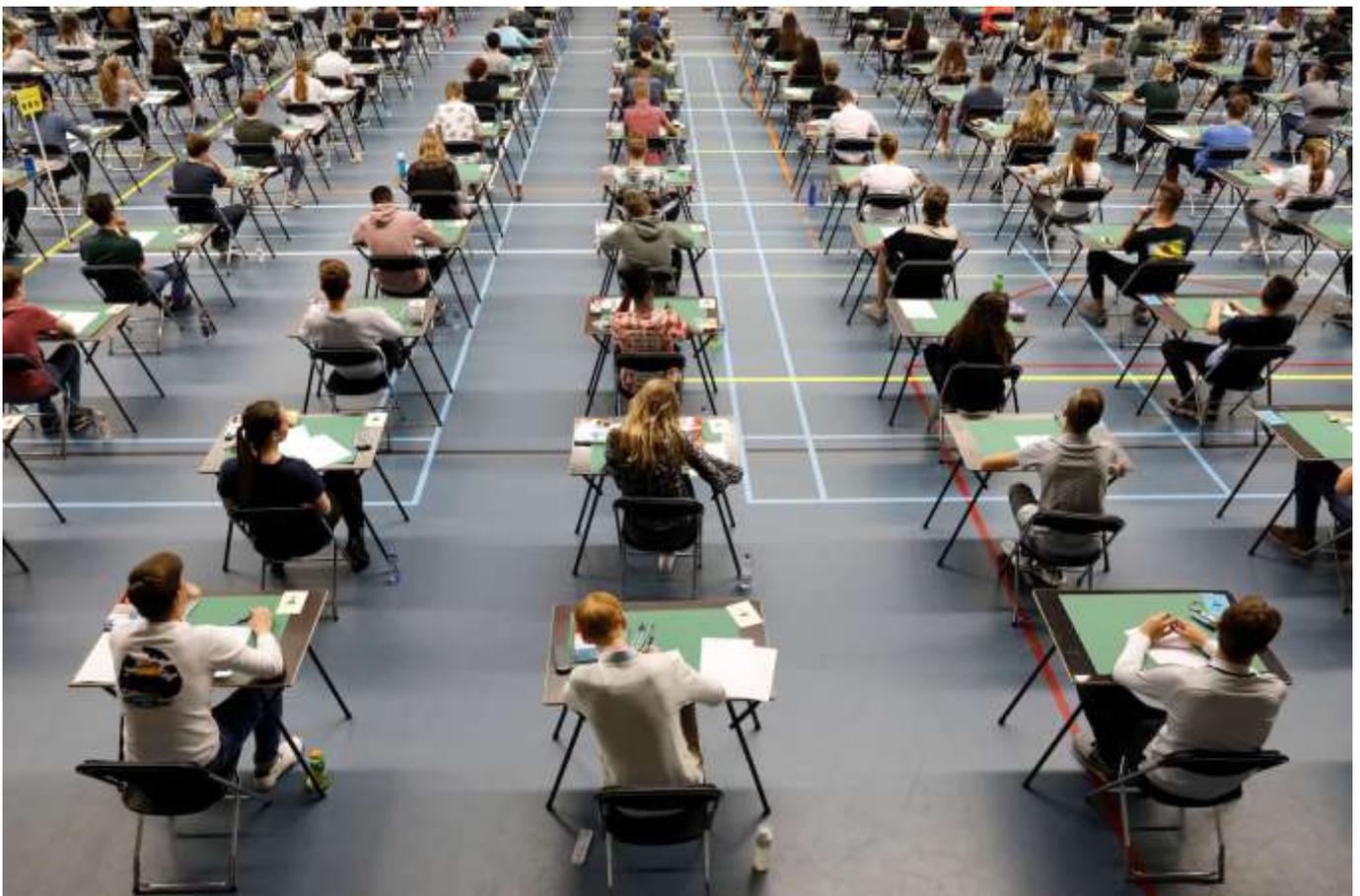


ALGORITHMIQUE

IMPRIMER DES NUMÉROS DE TABLE EN PILES

APL-AML



Public:

- Étudiants
- Débutants
- et pourquoi pas, enseignants

Contenu :

- spatialiser l'information
- les tables imbriquées
- la logique séquentielle
- la logique combinatoire
- la pédagogie LCP
 - la structure itérative
 - la structure alternative
 - la structure complexe

Devoir de vacances d'algorithmique (Force 6/7)

■ ■ ■ SOMMAIRE ■ ■ ■

- *Avant-propos*
- 1. *Imprimer des numéros de table en piles*
 - 1.1. *Simulation*
 - 1.2. *Algorigramme LCP*
 - 1.3. *Programme "post_itBDD.ace"*
 - 1.3.1. *Les tables de la BDD sollicitées par le programme "post_itBDD"*
 - 1.3.2. *Le Programme source "post_itBDD.ace"*
 - 1.3.3. *Le shell "post_itBDD"*
 - 1.3.4. *Le fichier "tmp.out"*
 - 1.3.5. *Les requêtes SQL*
- 2. *La pédagogie LCP*
 - 2.1. *Méthodologie "Logique de Construction de Programme" (LCP)*
 - 2.2. *LCP et le programme "post_itBDD.ace"*
 - 2.2.1. *Structuration des données de Sortie (FLS)*
 - 2.2.2. *Structuration des données d'entrée (FLE)*
 - 2.2.3. *Structuration du programme (PRG)*
 - 2.2.4. *Algorigramme*
 - 2.2.5. *Structuration in extenso du programme*
- 3. *Édition des numéros de table*
 - 3.1. *Le programme "post_itALL.ace"*
 - 3.2. *L'Algorigramme*
 - 3.3. *Le programme source "post_itALL.ace"*
 - 3.4. *Le shell post_itALL*
 - 3.5. *Exemple de numéros de table*
- 4. *Méthodologie de programmation LCP*

• **Motivations et raisons ayant poussé à étudier le sujet**

Trouver dans nos développements un sujet d'algorithmique original, simple, qui parle à tout le monde et susceptible de transmettre pédagogiquement la réalité de notre métier de développeur, n'a rien d'évident.

Nos développements sont tous très spécifiques en termes de problématique, de fonctionnalité ou de langage et donc peu propices à devenir un sujet pédagogique intéressant du point de vue de la conception, de l'algorithmique et de la programmation.

• **Présentation du sujet et du type de document**

Le sujet de ce billet, extrait d'une application Examens-Concours, est le seul parmi plus de 3.000 fonctionnalités développées à satisfaire tous les critères.

C'est en quelque sorte une mini application enchainant plusieurs fonctionnalités batch :

1. Une requête SQL renseigne l'attribut "**ec.places**" (dernier N° de place) de la table "**ec**" (Examens-Concours).

Le dernier numéro de place permettra de déterminer le nombre de pages : $(ec_places + 3) / 4$

2. Un programme (**post_itBDD.ace**) crée le fichier "**tmp.out**" à partir de l'attribut "**cec.place**" de la table "**cec**".

L'attribut "**tmp.post_it**" du fichier "**tmp.out**" est le critère de tri à renseigner pour permettre l'édition en piles des numéros de table.

3. Une première requête SQL charge le fichier "**tmp.out**" dans la table "**tmp**" créée pour ce genre de traitements batch.

4. Une deuxième requête SQL renseigne l'attribut "**cec.post_it**" de la table "**cec**" depuis la table "**tmp**".

5. Un programme (**post_itALL.ace**) imprime les numéros de table en piles.

Mon *Devoir de vacances* ne concerne que l'algorithme du premier programme (**post_itBDD.ace**) n'impliquant que deux attributs (ec.place et cec.place) :

Entités	Tables	Attributs	Libellés
Examens-Concours	ec	ec.cts char (9) not null, ec.places smallint,	Code-Type-Session ec Dernier N° de place
Candidatures Examens-Concours	cec	cec.cts char (9) not null, cec.n cec serial not null, cec.place smallint, cec.post_it smallint,	Code-Type-Session ec N° de candidature N° de place N° de post_it
Fichier "tmp.out" Table temporaire	tmp	tmp.cts char (9) not null, tmp.n cec serial not null, tmp.place smallint, tmp.post_it smallint,	Code-Type-Session ec N° de candidature N° de place N° de post_it

• **Précision de l'objectif poursuivi**

Posté le 18/07/2021 sur le [Forum Algorithmes et structures de données](#), mon *Devoir de vacances* n'avait pas pour objectif de solliciter l'aide des membres mais de les confronter à des conditions réelles de développement d'une problématique de gestion :

- ➡ **[Actualité] Sujet d'algorithmique original : imprimer des numéros de table en piles**

Dans le cadre d'un concours multiple, cette discussion a fait l'objet d'une **[Actualité]** sur le site :

- ➡ **[Actualité] Devoir de vacances d'algorithmique : imprimer des numéros de table en piles**

Ce devoir de vacances était à l'origine un billet de mon blog logique [ALGORITHMIQUE](#) :

- ➡ **[TUTORIEL] Imprimer des numéros de table en piles**

• **Difficultés rencontrées**

Incompréhension totale de la problématique des deux seuls intervenants en 20 jours.

1. Imprimer des numéros de table en piles

Juste pour vous faire partager le plus sympathique sujet d'algorithmique auquel j'ai été confronté.

Anecdote :

Nous sommes début mars 1992 et je développe l'application *examens-concours* depuis à peine plus d'un mois lorsqu'une gestionnaire vient s'installer en face de moi. Je la vois inscrire des chiffres avec un gros feutre noir sur des morceaux cartonnés découpés dans des intercalaires.

- *Vous faites quoi ?*
- *Je fais des numéros de table pour le concours qui va avoir lieu la semaine prochaine, j'ai besoin de calme, les collègues sont trop bruyantes.*
- *Mais c'est fastidieux, ce travail ! Vous ne faites quand même pas ça pour tous les concours ? Si ?... Pour le prochain concours, je résous votre problème...*

Mais bien sûr, il n'est pas question de remplacer l'inconvénient d'inscrire des numéros par l'inconvénient de les trier. L'idée est donc d'imprimer ces numéros de table, quatre par page mais en piles, c'est-à-dire que sous le numéro « un » il y a le « deux », puis le « trois », etc. Deux coups de massicot, on réunit les quatre piles, on met un élastique autour et le tour est joué.

Et tant qu'à faire, plutôt que de se contenter d'un numéro sur un quart de feuille, autant y mettre le libellé du concours, la date, le lieu, la salle, les noms-prénoms et la consigne « *Numéro de place à reporter sur votre copie* ».

Avant l'envoi des convocations à un concours, un placement des candidatures par ordre alphabétique affecte chronologiquement un numéro de place à chaque candidat et le numéro affecté au dernier candidat est mémorisé dans l'attribut "**ec.places**" de la table "**ec**" des Examens-Concours.

Pour pouvoir imprimer en piles 4 numéros de place par page, il faut nécessairement affecter un nouveau numéro (**cec.post_it**) à chaque candidature (**cec.place**) de façon à ce que les candidatures triées sur ce nouveau numéro permettent au programme d'édition d'imprimer sur chaque page les quatre numéros de place qui conviennent.

Facile de créer ces nouveaux numéros ! Il suffit de connaître le dernier numéro de place (**ec.places**), de diviser par quatre et de prendre en compte l'éventuel reste.

Oui, mais... Car il y a toujours un « Oui, mais » ! Certains candidats peuvent s'inscrire aux deux concours interne et externe. Avant la date d'examen, ils doivent obligatoirement se déterminer pour l'un des deux concours. Cela implique la suppression d'une candidature dans l'un des deux, ce qui génère fatalement une place libre. Mais comment traiter une candidature qui n'existe plus dans la table des candidatures ?

Je vous laisse réfléchir à l'algorithme de création de ces nouveaux numéros... si toutefois cette problématique vous inspire.

Bonnes vacances !

1.1. Simulation

Création d'un jeu d'essai avec 22 candidatures (sans places libres)

Chaque même quart de feuille des pages à imprimer constitue une pile.

Deux tableaux permettent de spatialiser mentalement l'édition pour comprendre comment le NUMÉRO DE PLACE interagit avec le NUMÉRO DE POST-IT :

- **Tableau de gauche**

La table des candidatures, triée sur le NUMÉRO DE PLACE, est traitée par le premier programme "**post_itBDD.ace**" afin de déterminer le rang des places (**Post-it**) qui permettra au deuxième programme d'imprimer en piles.

Ce tableau en trois dimensions (2 x 2 x n) représente l'impression de 4 numéros de table par page, "n" étant le nombre de pages, à savoir n = 6 pour ce jeu d'essai.

Deux compteurs **Pile** et **Page** modélisent les coordonnées spatiales - verticale et horizontale - des NUMÉROS DE PLACE et permettent de calculer **Post-it** pour chaque candidature :

- **Pile** s'incrément de 1 à 4
- **Page** s'incrémente de 1 à n (n étant égal à 6 pour ce jeu d'essai)

Pile et **Page** ne sont finalement que les indices de deux tables virtuelles.

Sont surlignées les lignes correspondant aux NUMÉROS DE PLACE (1, 7, 13 et 18) à imprimer sur la première page.

La réflexion consiste à s'intéresser aux lignes avec un même numéro de page pour comprendre comment évolue **Post-it** en fonction de **Pile** et **Page**.

- **Tableau de droite**

La table des candidatures, triée sur le NUMÉRO DE POST-IT, permet au deuxième programme "**post_itALL.ace**" d'imprimer les NUMÉROS DE PLACE en piles.

Sont surlignées les lignes correspondant aux NUMÉROS DE PLACE imprimés dans la première pile.

ec_places = Plus grand numéro de place = nombre de candidats = 22
 Nombre de pages = (ec_places + 3) / 4 = 6
 N° de post-it = N° de la pile + (N° de la page précédente * 4)
 => initialiser le numéro de page à zéro

TRI / NUMÉRO DE PLACE

Place	Post-it	Pile	Page
1	1	1	1
2	5	1	2
3	9	1	3
4	13	1	4
5	17	1	5
6	21	1	6
7	2	2	1
8	6	2	2
9	10	2	3
10	14	2	4
11	18	2	5
12	22	2	6
13	3	3	1
14	7	3	2
15	11	3	3
16	15	3	4
17	19	3	5
18	4	4	1
19	8	4	2
20	12	4	3
21	16	4	4
22	20	4	5

TRI / NUMÉRO DE POST-IT

Place	Post-it	Pile	Page
1	1	1	1
7	2	2	1
13	3	3	1
18	4	4	1
2	5	1	2
8	6	2	2
14	7	3	2
19	8	4	2
3	9	1	3
9	10	2	3
15	11	3	3
20	12	4	3
4	13	1	4
10	14	2	4
16	15	3	4
21	16	4	4
5	17	1	5
11	18	2	5
17	19	3	5
22	20	4	5
6	21	1	6
12	22	2	6

Exemple Place 9 : N° de la pile (2) + (N° de la page précédente (3-1) * 4) = N° de post-it (10)

Numéros de places

Le placement des candidats se réalise au dernier moment avant l'édition et l'envoi des convocations. Le dernier numéro de place correspond à ce moment-là au nombre d'inscrits mais ce nombre d'inscrits pourra évoluer entre le placement des candidats et la date du concours.

Les candidatures postérieures à la date limite d'inscription ne sont pas prises en compte (hors délais) mais une candidature émise avant la date limite d'inscription peut toujours être reçue accidentellement après le placement. L'application gère cette situation mais ce n'est pas le problème présentement. Pour être exhaustif, il y a également le cas des candidats dits COTOREP. Des places particulières doivent alors être attribuées à ces candidats si leur handicap nécessite un aménagement particulier. On leur attribue un numéro de place au-delà du dernier numéro de place. Mais c'est vraiment pour l'anecdote.

Places libres

Je n'ai pas concrétisé de place libre dans ma simulation mais c'est facile à faire. Il suffit par exemple de désister le candidat N° 13. Il faut alors supprimer **Place = 13** et **Post-it = 3**. On garde les informations **Pile = 3** et **Page = 1** pour visualiser notre simulation.

La candidature n'existant plus, le programme devra faire évoluer ces deux informations **Pile** et **Page** pour permettre de traiter correctement la candidature suivante.

Pour savoir s'il y a une place vide, le programme devra compter les candidatures traitées et comparer son compteur au numéro de place de la candidature en cours de traitement. Le programme devra évidemment prévoir le cas de plusieurs places vides qui se suivent, et qu'il y ait une ou plusieurs places vides au début comme à la fin de la table des candidatures à l'Examen-Concours.

Le programme d'édition quant-à lui va également s'apercevoir qu'il manque le **Post-it N° 3** et imprimera un numéro de table 13 avec « Place libre » comme nom-prénom.

Édition du jeu d'essai

Ci-dessous, les six pages de l'édition des numéros de table du jeu d'essai :

<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>1</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. AZNAVOUR Charles</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>7</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. BRUEL Patrick</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>2</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME BAEZ Joan</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>8</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. CABREL Francis</p>
<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>13</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME TUCK Hailey</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>18</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. LE FORESTIER Maxime</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>14</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. GOLMAN Jean-Jacques</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>19</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME PARADIS Vanessa</p>
<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>3</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. BASHUNG Alain</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>9</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. CLERC Julien</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>4</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME BIRKIN Jane</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>10</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. DELPECH Michel</p>
<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>15</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. HALLIDAY Johnny</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>20</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. REGGIANI Serge</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>16</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME JONASZ Michel</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>21</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. SHELLER William</p>
<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>5</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. BRASSENS Georges</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>11</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. DUTRONC Jacques</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>6</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. BREL Jacques</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>12</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME FARNER Hylène</p>
<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>17</p> <p>Numéro de place à reporter sur votre copie</p> <p>M. LAVILLIERS Bernard</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>22</p> <p>Numéro de place à reporter sur votre copie</p> <p>MME VARTAN Sylvie</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>PLACE LIBRE</p>	<p>CONCOURS INTERNE Salle : 1 SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p>PLACE LIBRE</p>

Cf. **3.5. Exemple de numéros de table**

1.2. Algorithme LCP

Lorsque la Logique de Construction de Programme est acquise, les structurations et l'algorithme restent à l'état de construction mentale quasi inconsciente.

Le programme "post_itBDD.ace" n'utilise que 20 instructions et gère deux compteurs que l'on peut assimiler à des tables (*voir la simulation ci-dessus*) dont le contenu est leur rang (*numéro*), autrement dit leur indice courant (*i_page* et *i_pile*) :

- une table des pages (nombre d'items = (dernier numéro de place + 3) / 4)
- et une table des piles de 4 items.

Si le programme a été écrit le 10/03/1992, son algorithme a été réalisé le 26/06/2019 pour créer ce Billet, soit 27 ans plus tard. Dans un souci pédagogique plus rigoureux, les structurations FLS, FLE et PRG ont quant à elles été réalisées encore plus récemment en juillet 2021, suite à la découverte sur internet d'un ancien support de cours LCP qui se transforme progressivement en tutoriel :

Logique de Construction des Programmes

Le programme

Le traitement consiste, à partir de chaque numéro de place et de ses numéros de pile et de page, à déterminer un nouveau numéro (*post_it*) de façon à ce que les candidatures triées sur ce nouveau numéro permettent d'imprimer les numéros de place en piles (4 numéros de place par page), sachant toutefois que certaines candidatures ont pu être supprimées après le placement suite à des désistements.

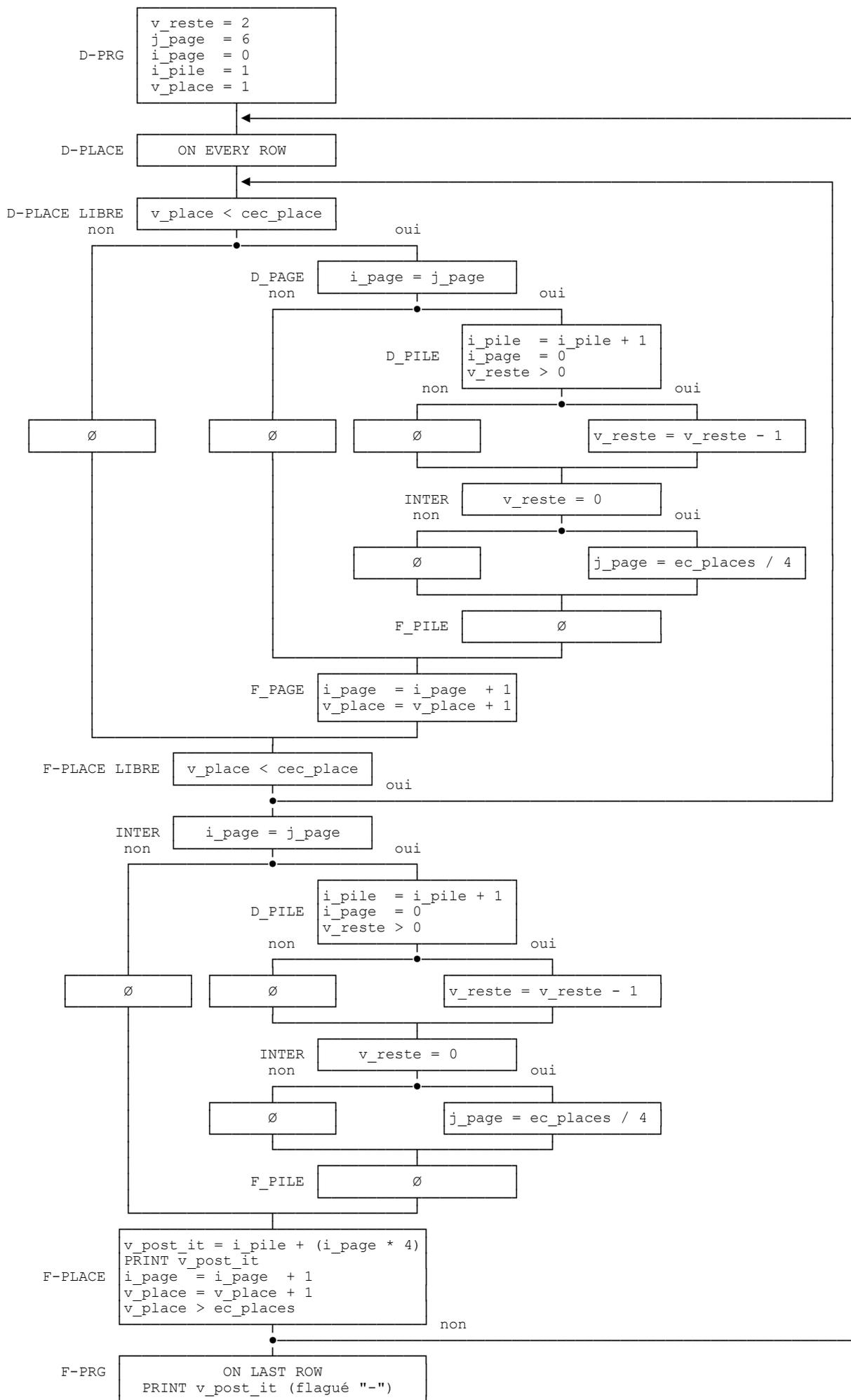
- **PLACE** : Le traitement *PLACE* correspond au *CANDIDAT*. Un numéro de place a été affecté aux candidats lors du placement par ordre alphabétique avant l'envoi des convocations.
- **PLACE LIBRE** : Le traitement *PLACE_LIBRE* palie en fait les éventuelles candidatures supprimées.
- **PILE** : *PILE*, coordonnée verticale, évolue de 1 à 4 en s'incrémentant de 1 après le traitement de la dernière place d'une *PILE*.
- **PAGE** : *PAGE*, coordonnée horizontale, s'incrémente de 1 à n, "n" étant le nombre de pages qu'il est prévu d'imprimer. *PAGE* est donc réinitialisé à 1 toutes les n places traitées.

Lors du traitement de la dernière place d'une *PILE*, *PAGE* peut valoir "n - 1" lorsque le nombre de places du concours ne correspond pas à un multiple de 4.

```
Numéro de post-it = Numéro de la pile + (Numéro de la page précédente * 4)
=> initialiser le numéro de page à zéro
```

Initialisations :

```
LET j_page = ec_places / 4          -> j_page = 22 / 4          = 5
LET v_reste = ec_places - (j_page * 4) -> v_reste = 22 - (5 * 4) = 2 -> v_reste = 2
LET j_page = (ec_places + 3) / 4    -> j_page = (22 + 3) / 4 = 6 -> j_page = 6
LET i_page = 0                      -> i_page = 0
LET i_pile = 1                       -> i_pile = 1
LET v_place = 1                     -> v_place = 1
```



1.3. Programme "post_itBDD.ace"

Cet exercice d'algorithmique n'est autre que le programme "**post_itBDD.ace**" d'une application Examens-Concours développée avec le SGBD Informix de 1992 à 2007.

Dans un environnement SGBD, c'est le programme qui constitue et structure le FLE dont il a besoin.

Fichier Logique d'Entrée (FLE)

- **ec** (examens-concours),
- **cec** (candidatures examens-concours),
- **In** (logname). **Fichier Logique de Sortie (FLS)**
- **tmp** (table temporaire)

Fichier Logique de Sortie (FLS)

- **tmp** (table temporaire)

1.3.1. Les tables de la BDD sollicitées par le programme "post_itBDD.ace"

- **Table ec**

Pour les besoins de l'édition des post_it, la table « ec » s'est enrichie de l'attribut « *ec.places* » correspondant à la dernière place, donc au nombre de candidats *ec.inscrits* au concours. Les attributs « *place* » et « *inscrits* » sont renseignés lors du placement des candidats, effectué une seule fois juste avant l'impression des convocations.

Pour différentes raisons, le nombre d'inscrits peut évoluer après le placement et l'envoi des convocations. Il peut exceptionnellement augmenter mais surtout diminuer suite à des désistements. Le placement ne pouvant plus être relancé, l'ajout éventuel d'une candidature peut toujours se faire mais manuellement par le DBA (DataBase Administrator). Un désistement a pour conséquences la suppression physique de la candidature et l'impression d'un post-it « Place libre » lors de l'édition des numéros de table (*cec.post-it*).

- **Table cec**

La table « cec » s'est enrichie de l'attribut « *post-it* » afin de pouvoir imprimer les numéros de table en piles. L'objectif du programme "**post_itBDD.ace**" consiste à associer un numéro de post-it à chaque numéro de place de façon à ce que les numéros de place puissent être imprimés quatre par page dans l'ordre des numéros de post-it (voir le jeu d'essai, ci-dessous).

- **Table In**

Chaque gestionnaire est identifié par son logname lors de sa connexion. Transmis en paramètre au programme, le logname permet de s'assurer que le terminaliste exécutant le programme est bien habilité à le faire.

Les attributs concernant le « suivi... » permettent de renseigner les coordonnées du gestionnaire-référent dans certains états comme les convocations.

Chaque bureau de gestionnaires est doté d'une imprimante HP Laser, mais le parc des imprimantes n'est pas strictement homogène. La table In permet d'adapter les impressions à l'imprimante de destination associée au gestionnaire.

- **Table tmp**

Cette table est utilisée dans l'application pour intervenir sur la table "**cec**" (candidatures) via une requête SQL.

Le fichier "**tmp.out**" créé par le programme "**post_itBDD.ace**" sera chargé dans la table "**tmp**" pour renseigner via une requête SQL l'attribut "**cec.post_it**" de la table "**cec**" (candidatures).

1.3.2. Le Programme source "post_itBDD.ace"

```
{===== post_itBDD =====}
{
{ ACE      : post_itBDD.ace      }
{
{ SHELL   : ../shell_1/post_itBDD }
{
{ AUTEUR  : APL-AML             }
{
{ Objet   : Création du fichier "${LOGNAME}.out" à loader dans la table tmp }
{         : pour mise à jour de l'information cec.post_it }
{         : à l'aide du sql "post_itBDD.sql" }
{
{         : L'édition des post_it s'effectue à l'aide du shell "post_itALL" }
{
{         : "post_itBDD.ace" et "post_itALL.ace" gèrent les places libres. }
{         : Il est donc possible de relancer éventuellement cette chaîne. }
{
{ Date    : 10 Mars 1992 }
{
{=====}

{=====} DATABASE concours END {=====}
{=====}

{=====} DEFINE {=====}
{}
{} PARAM[1] p_c_ec          CHAR(5)
{} PARAM[2] p_t_ec          CHAR(2)
{} PARAM[3] p_s_ec          CHAR(2)
{} PARAM[4] p_logname       CHAR(8)
{}
{} VARIABLE j_page         INTEGER
{} VARIABLE i_page         INTEGER
{} VARIABLE i_pile         INTEGER
{} VARIABLE v_post_it      INTEGER
{} VARIABLE v_place        INTEGER
{} VARIABLE v_reste        INTEGER
{}
{=====} END {=====}

{=====} INPUT =====
{}
{} PROMPT FOR v_dp USING " CONCOURS : "
{}
{=====} END =====

{=====} OUTPUT {=====}
{}
{} REPORT TO PRINTER
{} REPORT TO PIPE "cat > ${LOGNAME}.out"
{} TOP MARGIN 0
{} BOTTOM MARGIN 0
{} LEFT MARGIN 0
{} PAGE LENGTH 58
{}
{=====} END {=====}
```

```

{=====} SELECT {=====}

        ec.cts                ec_cts,
        ec.c_ec               ec_c_ec,
        ec.t_ec               ec_t_ec,
        ec.s_ec               ec_s_ec,
        ec.places             ec_places,

        cec.n_cec             cec_n_cec,
        cec.place             cec_place

FROM      ec,
         cec,
         ln

WHERE     ec.c_ec             = $p_c_ec
AND       ec.t_ec             = $p_t_ec
AND       ec.s_ec             = $p_s_ec

AND       ec.cts              = cec.cts

AND       (cec.c_decision     = "A"           OR cec.c_decision = "C")
AND       cec.place           IS NOT NULL

AND       ln.logname          = $p_logname
AND       (ln.service         IS NULL         OR ln.service     = ec.service)

{=====} ORDER BY {=====}

        ec_cts,
        cec_place

{=====} END {=====}

{=====} FORMAT {=====}

{-----} BEFORE GROUP OF ec_cts {-----}

LET j_page = ec_places / 4
LET v_reste = ec_places - (j_page * 4)
LET j_page = (ec_places + 3) / 4
LET i_page = 0
LET i_pile = 1
LET v_place = 1

{-----} FIRST PAGE HEADER {-----}
{-----} PAGE HEADER {-----}
{-----} PAGE TRAILER {-----}

```


1.3.3. Le shell "post_itBDD"

```
# post_itBDD

clear

LENGTH=`expr "$P" : '.*'`
if test "$LENGTH" -eq 11
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 10
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 9
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 8
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 7
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi

echo "                " $P_1 $P_2 $P_3
echo ""
echo " EDITION DES NUMEROS DE TABLE"
echo " "
echo " - Le lancement de cette édition suppose le PLACEMENT déjà effectué. "
echo " - L'édition gère les places vides "
echo " (candidatures supprimées après que le PLACEMENT ait été effectué) "
echo " - L'édition comprend deux étapes : "
echo " 1. Préparation de l'édition avec mise à jour de la base "
echo " 2. Edition proprement dite des numéros de table "
echo " "
echo " TRAITEMENT ? [ <— ]=OK, [Del]=Abandon "
echo " "
echo " "

cd ../shell_1

if test -f ${LOGNAME}.prt
then rm ${LOGNAME}.prt
fi

if test -f ${LOGNAME}.out
then rm ${LOGNAME}.out
fi

SMSO=`tput smso`
RMSO=`tput rmso`
```

```

if [ "$LENGTH" = 0 ]
then

    tput cup 0 0
    echo " CONCOURS : "
    tput cup 0 12
    echo "$SMSO" "$RMSO"
    tput cup 0 12
    read P_1
    if [ "$P_1" = "" ]
    then P_1=0
    fi

    tput cup 0 0
    echo " " "$P_1 $P_2 $P_3"
    tput cup 0 0
    echo " TYPE : "
    tput cup 0 12
    echo "$SMSO" "$RMSO"
    tput cup 0 12
    read P_2
    if [ "$P_2" = "" ]
    then P_2=0
    fi

    tput cup 0 0
    echo " " "$P_1 $P_2 $P_3"
    tput cup 0 0
    echo " SESSION : "
    tput cup 0 12
    echo "$SMSO" "$RMSO"
    tput cup 0 12
    read P_3
    if [ "$P_3" = "" ]
    then P_3=0
    fi
    echo ""

    tput cup 0 0
    echo " " "$P_1 $P_2 $P_3"
fi

# TRAITEMENT ?
tput cup 11 14
echo "$SMSO" "$RMSO"
tput cup 11 14
read REPONSE
tput cup 11 14
echo " "

tput cup 8 59
echo "en cours...\c"

sacego -q ../ace_1/post_itBDD $P_1 $P_2 $P_3 $LOGNAME

ed - "${LOGNAME}.out" << !
\$-1d
w
q
!

cp ../sql_1/post_itBDD.sed ${LOGNAME}.sql

ed - "${LOGNAME}.sql" << !
,s/LOGNAME/${LOGNAME}/g
,s/C_EC/$P_1/g
,s/T_EC/$P_2/g
,s/S_EC/$P_3/g
w
q
!

if [ "$DBLANG" = "" ]
then isql -s concours -qcre ${LOGNAME} > /dev/null
else isql -s concours -lceq ${LOGNAME} > /dev/null
fi

# post_itALL

tput cup 9 59

echo "en cours...\c"

echo "TERMINE ! "

```

1.3.4. Le fichier "tmp.out"

568||SASU|CI|07|-|-|-|||1||1|||
569||SASU|CI|07|-|-|-|||2||5|||
570||SASU|CI|07|-|-|-|||3||9|||
571||SASU|CI|07|-|-|-|||4||13|||
572||SASU|CI|07|-|-|-|||5||17|||
573||SASU|CI|07|-|-|-|||6||21|||
574||SASU|CI|07|-|-|-|||7||2|||
575||SASU|CI|07|-|-|-|||8||6|||
576||SASU|CI|07|-|-|-|||9||10|||
577||SASU|CI|07|-|-|-|||10||14|||
578||SASU|CI|07|-|-|-|||11||18|||
579||SASU|CI|07|-|-|-|||12||22|||
580||SASU|CI|07|-|-|-|||13||3|||
581||SASU|CI|07|-|-|-|||14||7|||
582||SASU|CI|07|-|-|-|||15||11|||
583||SASU|CI|07|-|-|-|||16||15|||
584||SASU|CI|07|-|-|-|||17||19|||
585||SASU|CI|07|-|-|-|||18||4|||
586||SASU|CI|07|-|-|-|||19||8|||
587||SASU|CI|07|-|-|-|||20||12|||
588||SASU|CI|07|-|-|-|||21||16|||
589||SASU|CI|07|-|-|-|||22||20|||

1.3.5. Les requêtes SQL

```
{ tmp      (temporaire) -----}

create table tmp
(
n_cec      integer not null,
n_pec      integer,
c_ec       char(5) not null,
t_ec       char(2) not null,
s_ec       char(2) not null,
groupe     char(1),
epreuve    char(1),
option     char(1),
partiel    char(1),
c_et       char(8),
salle      char(5),
place      smallint,
lettre_cle char(1),
post_it    smallint,
jury       smallint,
passage    smallint,
admis_1    char(1),
admis_2    char(1),
liste      char(1),
rang       smallint,
exaequo    char(1),
flag       char(1)
) ;

grant all      on tmp to public;

{-----}
{
post_itBDD.sed
}

{ Renseignement ec.places = dernière place du concours      }
{ La dernière place déterminera le nombre de pages = (ec.places + 3) / 4      }

lock table ec in exclusive mode;

update ec
set      ec.places = (select max(cec.place)
                      from      cec
                      where      cec.c_ec = "C_EC"
                      and        cec.t_ec = "T_EC"
                      and        cec.s_ec = "S_EC")

where ec.c_ec = "C_EC"
and   ec.t_ec = "T_EC"
and   ec.s_ec = "S_EC";

unlock table ec;

{-----}

{ Renseignement de cec.post_it depuis le fichier "LOGNAME.out"      }
{ créé par le programme "post_itBDD.ace"      }

lock table tmp in exclusive mode;

delete from tmp where tmp.n_cec is not null;

load from "LOGNAME.out" insert into tmp;

update cec
set      (cec.post_it)
        = ((select tmp.post_it
            from      tmp
            where cec.n_cec = tmp.n_cec))
where cec.c_ec = "C_EC"
and   cec.t_ec = "T_EC"
and   cec.s_ec = "S_EC";

{-----}
```

2.1. Méthodologie « Logique de Construction de Programme » (LCP)

LCP procède par structuration :

La structure hiérarchique d'un programme se déduisant logiquement de la structure des données utilisées (FLS/FLE), la construction du programme commence par la structuration des données qu'il doit produire en sortie.

Au-delà de la structuration hiérarchique du programme déterminée par la structure des données, le degré de décomposition algorithmique peut descendre jusqu'au niveau de l'instruction. Cela dépend de la capacité d'abstraction du programmeur. La seule règle qui régit la réflexion :

« *Penser hiérarchiquement par traitements, autrement dit : raisonner du plus global vers le plus détaillé, par décompositions hiérarchiques successives sur la base des actions et non sur celle des conditions* ».

■ Structuration du FLS, du FLE et du PRG

Concernant ce programme "*post_itBDD.ace*", la structuration classique LCP est rudimentaire puisqu'il s'agit de créer une seule information en sortie (*post_it*) à partir d'une seule information en entrée (*place*).

1. Structuration des données de sortie (*informations attendues*).

Les données de sortie ne sont pas exclusivement des données destinées à être imprimées. Ce peut être des données de la BDD (écran de saisie) ou le résultat d'un traitement particulier comme c'est le cas pour le programme "*post_itBDD.ace*" proposé.

2. Structuration des données d'entrée (*informations utiles*).

LCP est une méthode de programmation conçue à l'origine dans un contexte batch où le **F**ichier **L**ogique en **E**ntree (**FLE**) était constitué en amont du programme. Dans un contexte BDD, c'est le développeur qui constitue et structure le **FLE** dont le programme a besoin. C'est la conception et la structuration préalable du **FLS** qui inspire la conception et la structuration du **FLE**.

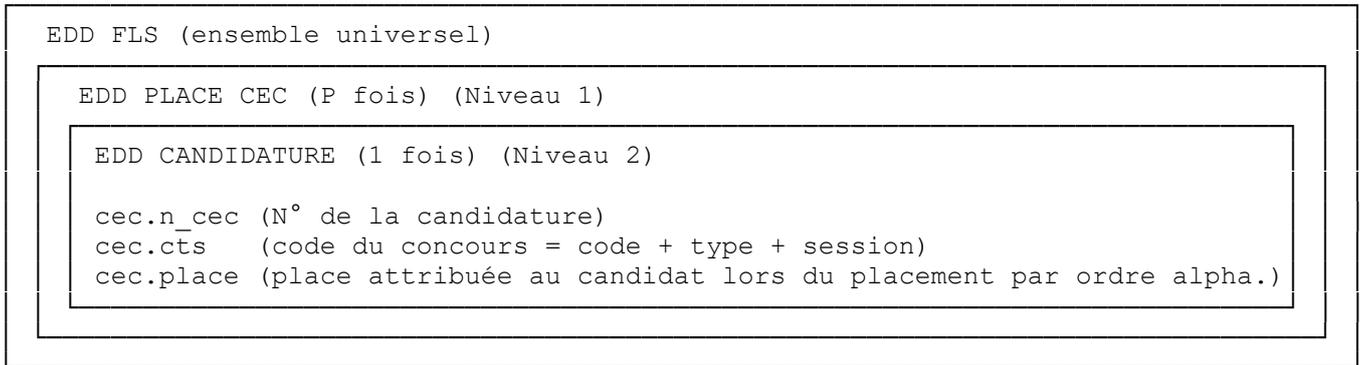
3. Structuration du programme.

La structuration du programme se déduit de celle du FLE pour ce qui concerne les **E**nsembles **D**e **D**onnées (**EDD**), mais la décomposition structurelle du programme peut devoir être poussée plus finement, au-delà des **EDD**, tout en respectant la logique LCP qui consiste à penser par traitements. C'est le cas pour ce programme "*post_itBDD.ace*".

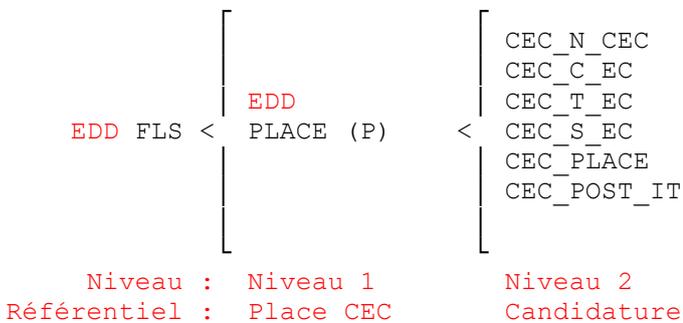
2.2. LCP et le programme "post_itBDD.ace"

2.2.1. Structuration des données de Sortie (FLS)"

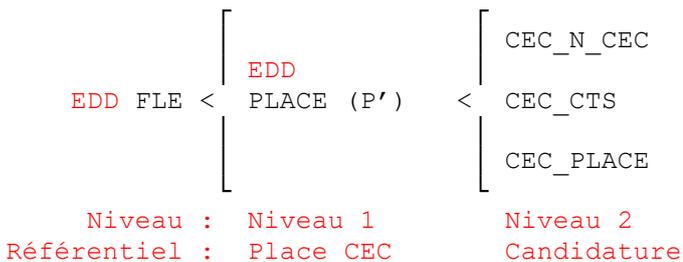
Représentation des Ensembles De Données (EDD) sous forme de patatoïdes :



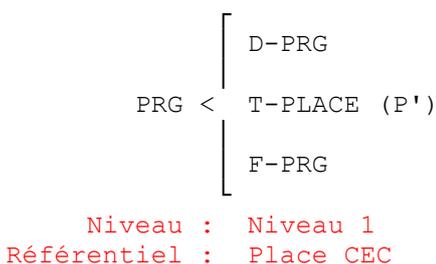
Représentation de ces mêmes Ensembles De Données à l'aide d'un diagramme avec des accolades :



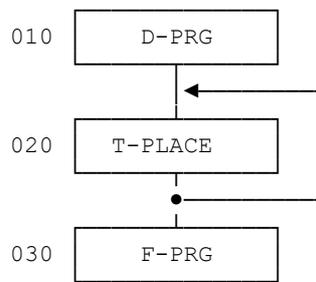
2.2.2. Structuration des données d'Entrée (FLE)



2.2.3. Structuration du Programme (PRG)



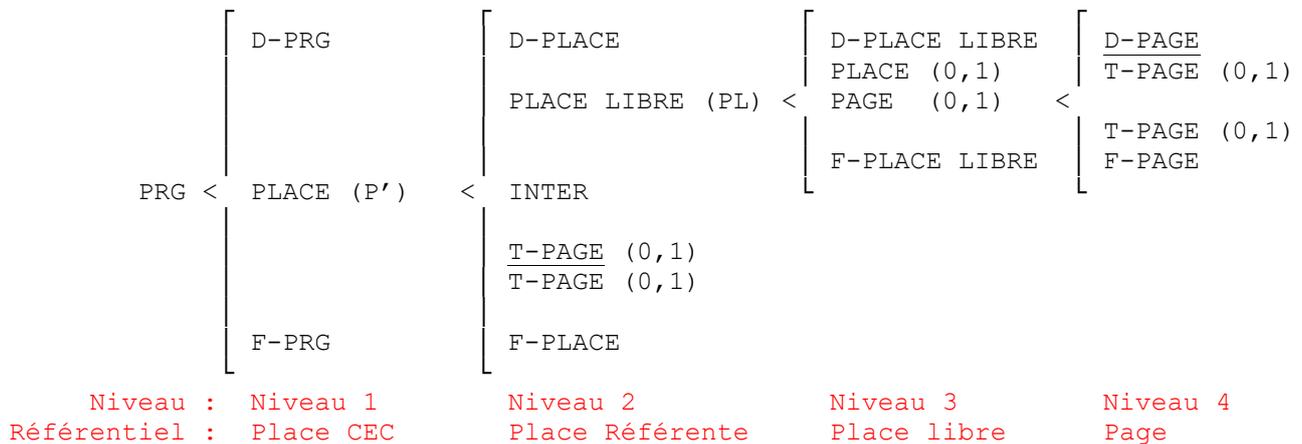
2.2.4. Algorithme



2.2.5. Structuration in extenso du Programme

Les candidatures supprimées n'apparaissent pas bien sûr dans le FLE. La structuration du programme prenant en compte les places libres générées par la suppression possible de certaines candidatures ne peut se réaliser que par la réflexion, tout comme la détermination du numéro de `post_it`. Symboliser cette réflexion sous forme d'un diagramme avec des accolades peut s'avérer très vite compliqué.

Traduction d'un début de cette réflexion sous forme d'un diagramme avec des accolades :



3. Édition des numéros de table

3.1. Le Programme "post_itALL.ace"

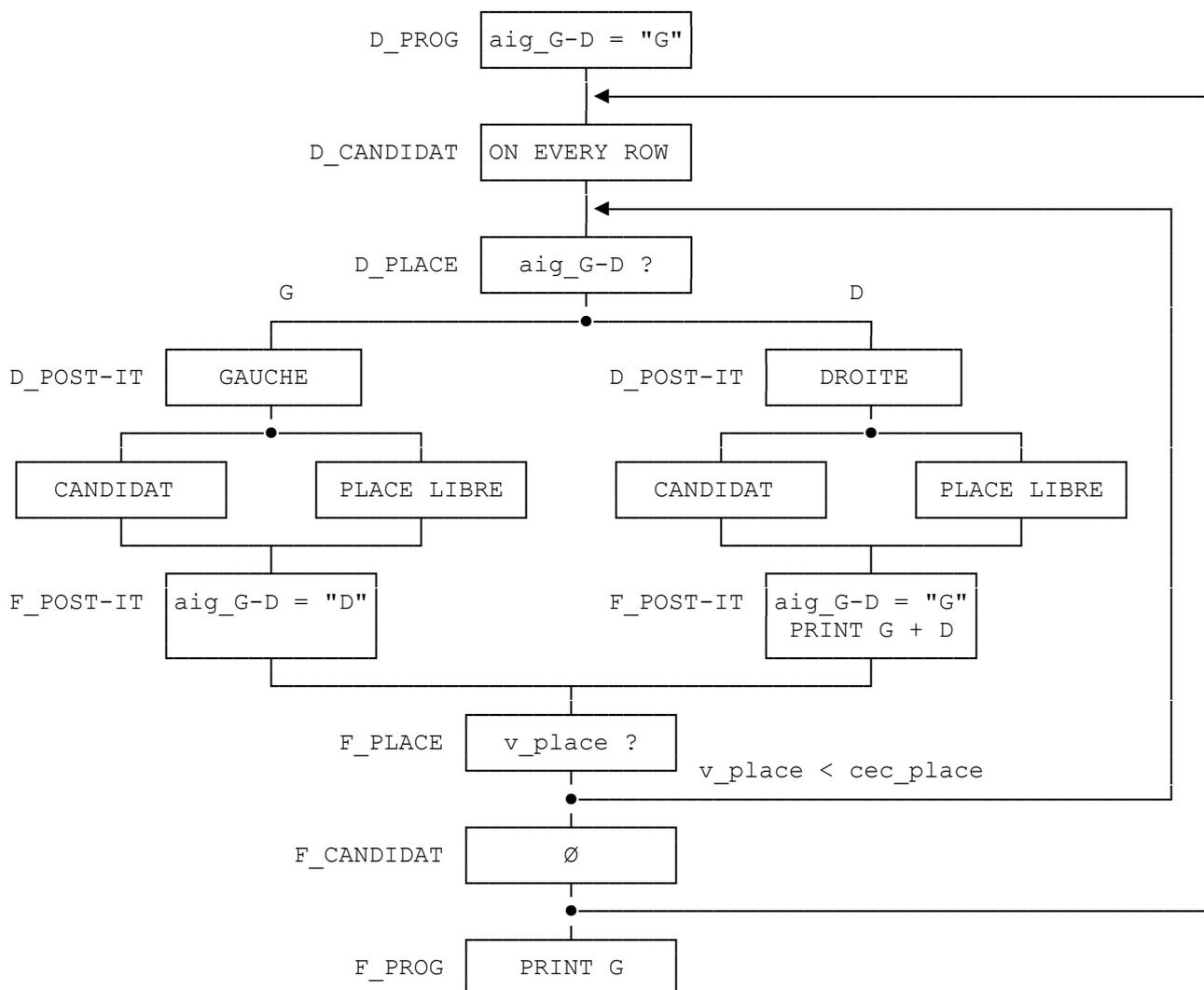
Le programme imprime l'ensemble des POST-IT (numéros de table). On retrouve évidemment le problème des places libres au moment de l'impression. Facile !

Comme pour le programme précédent, l'algorithme a été réalisé à posteriori, le 07/07/2019 pour décrypter le raisonnement par traitements réalisé au moment de la conception. Si l'algorithme paraît simple, la programmation est assez technique.

L'idée est d'imprimer les numéros de table, deux par deux. Pour simplifier la programmation, les deux numéros de table sont traités de la même façon. L'impression se réalise après le traitement du deuxième numéro de table.

La séquence logique « F_PROG » doit prévoir l'éventualité d'un dernier numéro de table à imprimer lorsque le nombre total de numéros de tables est impair.

3.2. L'algorithme



3.3. Le Programme source "post_itALL.ace"

```
{===== post_itALL =====}
{
{ ACE      : post_itALL.ace
{ SHELL   : ../shell_1/post_itALL
{
{ AUTEUR  : APL-APL
{
{ Objet   : Edition de l'ensemble des post_it (Version Ex&Co)
{
{ Date    : 10 Mars 1992
{
{=====}

{=====} DATABASE concours END {=====}
{=====}

{=====} DEFINE {=====}
{}
{} PARAM[1] p_c_ec          CHAR(5)
{} PARAM[2] p_t_ec          CHAR(2)
{} PARAM[3] p_s_ec          CHAR(2)
{} PARAM[4] p_logname       CHAR(8)
{}
{} VARIABLE i_loop         INTEGER
{} VARIABLE i_1             INTEGER
{ VARIABLE i_2             INTEGER
{ VARIABLE i_3             INTEGER
{ VARIABLE i_4             INTEGER
{ VARIABLE i_etiquette     INTEGER
{}
{} VARIABLE aig_pile        INTEGER
{} VARIABLE i_pile_0        INTEGER
{} VARIABLE i_pile_1        INTEGER
{} VARIABLE i_pile_2        INTEGER
{} VARIABLE v_place        INTEGER
{} VARIABLE v_reste        INTEGER
{}
{} VARIABLE v_post_it      INTEGER
{}
{} VARIABLE v_salle_0       CHAR(4)
{} VARIABLE v_salle_1       CHAR(4)
{} VARIABLE v_salle_2       CHAR(4)
{} VARIABLE v_salle_3       CHAR(4)
{}
{} VARIABLE v_hall_0        CHAR(38)
{} VARIABLE v_hall_1        CHAR(38)
{} VARIABLE v_hall_2        CHAR(38)
{} VARIABLE v_hall_3        CHAR(38)
{}
{} VARIABLE v1_salle        CHAR(5)
{} VARIABLE v1_hall         CHAR(38)
{} VARIABLE v1_place        INTEGER
{} VARIABLE v1_lettre_cle   CHAR(1)
{} VARIABLE v1_civilite     CHAR(3)
{} VARIABLE v1_nom_usage    CHAR(20)
{} VARIABLE v1_nom_naiss    CHAR(20)
{} VARIABLE v1_prenom       CHAR(20)
{}
{} VARIABLE v2_salle        CHAR(5)
{} VARIABLE v2_hall         CHAR(38)
{} VARIABLE v2_place        INTEGER
{} VARIABLE v2_lettre_cle   CHAR(1)
{} VARIABLE v2_civilite     CHAR(3)
{} VARIABLE v2_nom_usage    CHAR(20)
{} VARIABLE v2_nom_naiss    CHAR(20)
{} VARIABLE v2_prenom       CHAR(20)
{}
{} VARIABLE ctr_place       INTEGER
{}
{} VARIABLE ESC             CHAR(1)
{}
{=====} END {=====}
```

```

{===== INPUT =====}
{}
{} PROMPT FOR v_dp USING " CONCOURS : "
{}
{}
{===== END =====}

{=====} OUTPUT {=====}
{}
{} REPORT TO PRINTER
{} REPORT TO PIPE "cat > ${LOGNAME}.prt"
{} TOP MARGIN 0
{} BOTTOM MARGIN 0
{} LEFT MARGIN 0
{} PAGE LENGTH 80
{} TOP OF PAGE "^L"
{}
{=====} END {=====}

{=====} SELECT {=====}

    ec.cts          ec_cts,
    ec.c_ec         ec_c_ec,
    ec.t_ec         ec_t_ec,
    ec.s_ec         ec_s_ec,
    ec.l_ec_1       ec_l_ec_1,
    ec.l_ec_2       ec_l_ec_2,
    ec.placés       ec_places,

    tec.t_ec_l_maj  tec_t_ec_l_maj,

    sp.hall         sp_hall,

    cec.n_pec       cec_n_pec,
    cec.c_et        cec_c_et,
    cec.salle       cec_salle,
    cec.place       cec_place,
    cec.lettre_cle  cec_lettre_cle,
    cec.post_it     cec_post_it,

    pec.civilite    pec_civilite,
    pec.nom_naiss   pec_nom_naiss,
    pec.nom_usage   pec_nom_usage,
    pec.prenom      pec_prenom,
    pec.prenom_2eme pec_prenom_2eme,

    ln.pre_imprime  ln_pre_imprime,
    ln.papier_blanc ln_papier_blanc

FROM    ec,
        sp,
        tec,
        cec,
        pec,
        ln

WHERE   ec.c_ec      = $p_c_ec
AND     ec.t_ec      = $p_t_ec
AND     ec.s_ec      = $p_s_ec

AND     ec.t_ec      = tec.t_ec
AND     ec.cts       = sp.cts
AND     ec.cts       = cec.cts

AND     (cec.c_decision = "A"      OR cec.c_decision = "C")
AND     cec.c_et      = sp.c_et
AND     cec.salle     = sp.salle
AND     cec.n_pec     = pec.n_pec
AND     cec.post_it   IS NOT NULL

AND     ln.logname    = $p_logname
AND     (ln.service   IS NULL      OR ln.service   = ec.service)

{=====} ORDER BY {=====}

    ec_cts,
    cec_post_it

{=====} END {=====}

```

```

{=====} FORMAT {=====}

{-----} BEFORE GROUP OF ec_cts {-----}

LET v_post_it = 0
LET i_1 = 0

LET i_pile_0 = ec_places / 4
LET i_pile_1 = ec_places / 4
LET i_pile_2 = ec_places / 4
LET v_reste = ec_places - (i_pile_0 * 4)

IF v_reste > 0
THEN BEGIN
    LET i_pile_0 = i_pile_0 + 1
    LET v_reste = v_reste - 1
    END

IF v_reste > 0
THEN BEGIN
    LET i_pile_1 = i_pile_1 + 1
    LET v_reste = v_reste - 1
    END

IF v_reste > 0
THEN BEGIN
    LET i_pile_2 = i_pile_2 + 1
    LET v_reste = v_reste - 1
    END

LET aig_pile = 3

{----- FIRST PAGE HEADER -----}

LET ESC = ASCII 027

PRINT ESC, "E", ESC, "&k0g4S", ESC, "&l1o76p1x6D",
    ESC, ln_papier_blanc, ESC, "&a3L", ESC, "(10U";

{----- PAGE HEADER -----}

{----- PAGE TRAILER -----}

```

```

{-----} ON EVERY ROW {-----}

LET v_post_it = v_post_it + 1

FOR i_loop = v_post_it TO cec_post_it
DO BEGIN

IF i_1 = 0
THEN BEGIN

    LET i_1      = 1

    IF aig_pile = 1
    THEN LET aig_pile = 2
    ELSE LET aig_pile = 0

    IF v_post_it = cec_post_it
    THEN BEGIN
        IF aig_pile = 0
        THEN BEGIN
            LET v_salle_0 = cec_salle
            LET v_hall_0  = sp_hall
            END
        ELSE BEGIN
            LET v_salle_2 = cec_salle
            LET v_hall_2  = sp_hall
            END

        LET v1_salle      = cec_salle
        LET v1_hall       = sp_hall
        LET v1_place      = cec_place
        LET v1_lettre_cle = cec_lettre_cle
        LET v1_civilite   = pec_civilite
        LET v1_nom_usage  = pec_nom_usage

        IF pec_civilite = "MME"
        AND pec_nom_naiss IS NOT NULL
        AND pec_nom_naiss <> pec_nom_usage

        THEN LET v1_nom_naiss = pec_nom_naiss
        ELSE LET v1_nom_naiss = ""

        LET v1_prenom = pec_prenom CLIPPED
    {
    LET v1_prenom = pec_prenom CLIPPED, 1 SPACE, pec_prenom_2eme CLIPPED
    }
    END

ELSE BEGIN

    IF aig_pile = 0
    THEN BEGIN
        LET v1_salle = v_salle_0
        LET v1_hall  = v_hall_0
        END
    ELSE BEGIN
        LET v1_salle = v_salle_2
        LET v1_hall  = v_hall_2
        END

    IF aig_pile = 0
    THEN LET v1_place = PAGENO
    ELSE LET v1_place = i_pile_0 + i_pile_1 + PAGENO

    LET v1_lettre_cle = ""
    LET v1_civilite   = ""
    LET v1_nom_usage  = ""
    LET v1_nom_naiss  = ""
    LET v1_prenom     = ""
    LET v_post_it     = v_post_it + 1

    END

END

END

```

```

ELSE BEGIN

    LET i_1          = 0

    IF aig_pile = 0
    THEN LET aig_pile = 1
    ELSE LET aig_pile = 3

    IF v_post_it = cec_post_it
    THEN BEGIN
        IF aig_pile = 1
        THEN BEGIN
            LET v_salle_1 = cec_salle
            LET v_hall_1  = sp_hall
            END
        ELSE BEGIN
            LET v_salle_3 = cec_salle
            LET v_hall_3  = sp_hall
            END

        LET v2_salle      = cec_salle
        LET v2_hall       = sp_hall
        LET v2_place      = cec_place
        LET v2_lettre_cle = cec_lettre_cle
        LET v2_civilite   = pec_civilite
        LET v2_nom_usage  = pec_nom_usage

        IF pec_civilite = "MME"
        AND pec_nom_naiss IS NOT NULL
        AND pec_nom_naiss <> pec_nom_usage
        THEN LET v2_nom_naiss = pec_nom_naiss
        ELSE LET v2_nom_naiss = ""

        LET v2_prenom = pec_prenom CLIPPED
    {
        LET v2_prenom = pec_prenom CLIPPED, 1 SPACE, pec_prenom_2eme CLIPPED
    }
        END

    ELSE BEGIN
        IF aig_pile = 1
        THEN BEGIN
            LET v2_salle = v_salle_1
            LET v2_hall  = v_hall_1
            END
        ELSE BEGIN
            LET v2_salle = v_salle_3
            LET v2_hall  = v_hall_3
            END

        LET v2_salle = cec_salle
        LET v2_hall  = sp_hall

        IF aig_pile = 1
        THEN LET v2_place = i_pile_0 + PAGENO
        ELSE LET v2_place = i_pile_0 + i_pile_1 + i_pile_2 + PAGENO

        LET v2_lettre_cle = ""
        LET v2_civilite   = ""
        LET v2_nom_usage  = ""
        LET v2_nom_naiss  = ""
        LET v2_prenom     = ""
        LET v_post_it     = v_post_it + 1

        END
    END

```

```

IF sp_hall IS NULL
OR sp_hall = " "
THEN BEGIN

    PRINT ESC, "&k4S", ESC, "(s3B",
           tec_t_ec_l_maj, 16 SPACES,
           "Salle :", v1_salle,
           12 SPACES,
           tec_t_ec_l_maj, 16 SPACES,
           "Salle :", v2_salle, ESC, "(s0B"

    SKIP 1 LINE

    PRINT ESC, "&f0S", ESC, "(s1p14v3b4101T",
           ec_l_ec_1, ESC, "(s0p10v0b3T",
           ESC, "&f1S", 71 SPACES,
           ESC, "(s1p14v3b4101T",
           ec_l_ec_1, ESC, "(s0p10v0b3T"

    PRINT ESC, "&f0S", ESC, "(s1p14v3b4101T",
           ec_l_ec_2, ESC, "(s0p10v0b3T",
           ESC, "&f1S", 71 SPACES,
           ESC, "(s1p14v3b4101T",
           ec_l_ec_2, ESC, "(s0p10v0b3T"

    SKIP 5 LINES

    END

ELSE BEGIN

    PRINT ESC, "(s3B", v1_hall, 33 SPACES, v2_hall, ESC, "(s0B"

    SKIP 1 LINE

    PRINT ESC, "&k4S", ESC, "(s3B",
           tec_t_ec_l_maj, 41 SPACES,
           tec_t_ec_l_maj CLIPPED, ESC, "(s0B"

    SKIP 1 LINE

    PRINT ESC, "&f0S", ESC, "(s1p14v3b4101T",
           ec_l_ec_1, ESC, "(s0p10v0b3T",
           ESC, "&f1S", 71 SPACES,
           ESC, "(s1p14v3b4101T",
           ec_l_ec_1, ESC, "(s0p10v0b3T"

    PRINT ESC, "&f0S", ESC, "(s1p14v3b4101T",
           ec_l_ec_2, ESC, "(s0p10v0b3T",
           ESC, "&f1S", 71 SPACES,
           ESC, "(s1p14v3b4101T",
           ec_l_ec_2, ESC, "(s0p10v0b3T"

    SKIP 3 LINES

    END

PRINT ESC, "&f0S";

```

```

IF v1_lettre_cle IS NULL
OR v1_lettre_cle = " "
THEN BEGIN
  IF v1_place < 1000 THEN PRINT 2 SPACES;
  IF v1_place < 100 THEN PRINT 2 SPACES;
  IF v1_place < 10 THEN PRINT 2 SPACES;
  PRINT 22 SPACES, ESC, "(slp50v3b4101T",
    v1_place USING "<<<< ",
    ESC, "(s0p10v0b3T",
    ESC, "&f1S", 71 SPACES;
  END
ELSE PRINT 15 SPACES, ESC, "(slp50v3b4101T",
  v1_place USING "&&&& ",
  ESC, "*v0n1O", ESC, "*v1T",
  ESC, "(slp50v32s3b4101T",
  ESC, "*c45G", ESC, "*v2T", v1_lettre_cle,
  ESC, "(s0p10v0b3T", ESC, "*v0T",
  ESC, "&f1S", 71 SPACES;

IF v2_lettre_cle IS NULL
OR v2_lettre_cle = " "
THEN BEGIN
  IF v2_place < 1000 THEN PRINT 2 SPACES;
  IF v2_place < 100 THEN PRINT 2 SPACES;
  IF v2_place < 10 THEN PRINT 2 SPACES;
  PRINT 22 SPACES, ESC, "(slp50v3b4101T",
    v2_place USING "<<<< ",
    ESC, "(s0p10v0b3T"
  END
ELSE PRINT 15 SPACES, ESC, "(slp50v3b4101T",
  v2_place USING "&&&& ",
  ESC, "*v0n1O", ESC, "*v1T",
  ESC, "(slp50v32s3b4101T",
  ESC, "*c45G", ESC, "*v2T", v2_lettre_cle,
  ESC, "(s0p10v0b3T", ESC, "*v0T"

PRINT ESC, "&f0S",
  ESC, "*c1465a80b15g2P", ESC, "*v0n1O",
  ESC, "&f1S", 71 SPACES,
  ESC, "*c1465a80b15g2P", ESC, "*v0n1O"

IF v1_nom_usage IS NOT NULL
AND ec_t_ec <> "EB"
THEN PRINT ESC, "&f0S", ESC, "&k4S",
  "      Numéro de place à reporter sur votre copie";
ELSE
IF ec_t_ec = "EB"
THEN PRINT ESC, "&f0S", ESC, "&k4S";
ELSE PRINT ESC, "&f0S", ESC, "&k4S", 24 SPACES, "Place libre";

IF v2_nom_usage IS NOT NULL
AND ec_t_ec <> "EB"
THEN PRINT ESC, "&f1S", ESC, "&k4S", 71 SPACES,
  "      Numéro de place à reporter sur votre copie"
ELSE
IF ec_t_ec = "EB"
THEN PRINT ESC, "&f1S", ESC, "&k4S";
ELSE PRINT ESC, "&f1S", ESC, "&k4S", 93 SPACES, "Place libre";

SKIP 3 LINES

```

```

IF v1_nom_naiss IS NULL
THEN IF ec_t ec = "EB"
    THEN PRINT ESC, "&f0S", ESC, "(slp14v3b4101T",
        v1_nom_usage CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T";
    ELSE PRINT ESC, "&f0S", ESC, "(slp14v3b4101T",
        v1_civilite, 1 SPACE, v1_nom_usage CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T";

IF v1_nom_naiss IS NOT NULL
THEN IF ec_t ec = "EB"
    THEN PRINT ESC, "&f0S", ESC, "(slp14v3b4101T",
        v1_nom_naiss CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T";
    ELSE PRINT ESC, "&f0S", ESC, "(slp14v3b4101T",
        v1_civilite, 1 SPACE, v1_nom_naiss CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T";

IF v1_nom_usage IS NULL
THEN PRINT ESC, "&f0S";

IF v2_nom_naiss IS NULL
THEN IF ec_t ec = "EB"
    THEN PRINT ESC, "&f1S", 71 SPACES,
        ESC, "(slp14v3b4101T",
        v2_nom_usage CLIPPED,
        1 SPACE, ESC, "(s0B", v2_prenom CLIPPED,
        ESC, "(s0p10v0b3T"
    ELSE PRINT ESC, "&f1S", 71 SPACES,
        ESC, "(slp14v3b4101T",
        v2_civilite, 1 SPACE, v2_nom_usage CLIPPED,
        1 SPACE, ESC, "(s0B", v2_prenom CLIPPED,
        ESC, "(s0p10v0b3T"

IF v2_nom_naiss IS NOT NULL
THEN IF ec_t ec = "EB"
    THEN PRINT ESC, "&f1S", 71 SPACES,
        ESC, "(slp14v3b4101T",
        v2_civilite, 1 SPACE, v2_nom_naiss CLIPPED,
        1 SPACE, ESC, "(s0B", v2_prenom CLIPPED,
        ESC, "(s0p10v0b3T"
    ELSE PRINT ESC, "&f1S", 71 SPACES,
        ESC, "(slp14v3b4101T",
        v2_civilite, 1 SPACE, v2_nom_naiss CLIPPED,
        1 SPACE, ESC, "(s0B", v2_prenom CLIPPED,
        ESC, "(s0p10v0b3T"

IF v2_nom_usage IS NULL
THEN PRINT ESC, "&f1S", 1 SPACE

IF v1_nom_naiss IS NOT NULL
THEN PRINT ESC, "&f0S", "pouse ",
    ESC, "(slp14v3b4101T", v1_nom_usage CLIPPED, ESC, "(s0p10v0b3T";
ELSE PRINT ESC, "&f0S";

IF v2_nom_naiss IS NOT NULL
THEN PRINT ESC, "&f1S", 71 SPACES, "pouse ",
    ESC, "(slp14v3b4101T", v2_nom_usage CLIPPED, ESC, "(s0p10v0b3T"
ELSE PRINT ESC, "&f1S"

IF LINENO < 25
THEN SKIP 8 LINES
ELSE SKIP TO TOP OF PAGE

END

END

```

```

{----- TRAITEMENT FIN DE PAGE -----}
{----- TRAITEMENT EN-TETE DE PAGE -----}
{----- TRAITEMENT LIGNE COURANTE -----}
{----- AFTER GROUP OF ec_cts -----}
{-----} ON LAST ROW {-----}

IF i_1 = 1
THEN BEGIN
  IF sp_hall IS NULL
  OR sp_hall = " "
  THEN BEGIN

    PRINT ESC, "(s3B",
      tec_t_ec_l_maj, 16 SPACES,
      "Salle :", v1_salle

    PRINT ESC, "(s0B"

    PRINT ESC, "(s1p14v3b4101T", ec_l_ec_1, ESC, "(s0p10v0b3T"
    PRINT ESC, "(s1p14v3b4101T", ec_l_ec_2, ESC, "(s0p10v0b3T"

    SKIP 5 LINES

    END

  ELSE BEGIN

    PRINT ESC, "(s3B", v1_hall

    SKIP 1 LINE

    PRINT tec_t_ec_l_maj CLIPPED
    PRINT ESC, "(s0B"
    PRINT ESC, "(s1p14v3b4101T", ec_l_ec_1, ESC, "(s0p10v0b3T"
    PRINT ESC, "(s1p14v3b4101T", ec_l_ec_2, ESC, "(s0p10v0b3T"

    SKIP 3 LINES

    END

  IF v1_lettre_cle IS NULL
  OR v1_lettre_cle = " "
  THEN BEGIN
    IF v1_place < 1000 THEN PRINT 2 SPACES;
    IF v1_place < 100 THEN PRINT 2 SPACES;
    IF v1_place < 10 THEN PRINT 2 SPACES;
    PRINT 22 SPACES, ESC, "(s1p50v3b4101T",
      v1_place USING "<<<< ",
      ESC, "(s0p10v0b3T"
    END
  ELSE PRINT 15 SPACES, ESC, "(s1p50v3b4101T",
    v1_place USING "&&&& ",
    ESC, "*v0n10", ESC, "*v1T",
    ESC, "(s1p50v32s3b4101T",
    ESC, "*c45G", ESC, "*v2T", v1_lettre_cle,
    ESC, "(s0p10v0b3T", ESC, "*v0T"

    PRINT ESC, "*c1465a80b15g2P", ESC, "*v0n10"

    IF v1_nom_usage IS NOT NULL
    AND ec_t_ec <> "EB"
    THEN PRINT ESC, "&k4S",
      "          Numéro de place à reporter sur votre copie"

    SKIP 3 LINES

```

```

IF v1_civilite = "MME"
THEN IF ec_t_ec = "EB"
    THEN PRINT ESC, "(s1p14v3b4101T",
        v1_nom_naiss CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T"
    ELSE PRINT ESC, "(s1p14v3b4101T",
        v1_civilite, 1 SPACE, v1_nom_naiss CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T"
ELSE IF ec_t_ec = "EB"
    THEN PRINT ESC, "(s1p14v3b4101T",
        v1_nom_usage CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T"
    ELSE PRINT ESC, "(s1p14v3b4101T",
        v1_civilite, 1 SPACE, v1_nom_usage CLIPPED,
        1 SPACE, ESC, "(s0B", v1_prenom CLIPPED,
        ESC, "(s0p10v0b3T"

```

```

IF v1_civilite = "MME"
AND v1_nom_naiss IS NOT NULL
AND v1_nom_naiss <> v1_nom_usage
THEN PRINT "pouse ",
    ESC, "(s1p14v3b4101T",
    v1_nom_usage CLIPPED,
    ESC, "(s0p10v0b3T";
ELSE PRINT " ";

```

SKIP 8 LINES

END

```

{ - Reinitialisation de l'imprimante "HP LASERJET III Si" }
{=====} END {=====}

```

3.4. Le shell post_itALL

```
# post_itALL

clear

LENGTH=`expr "$P" : '.*'`
if test "$LENGTH" -eq 11
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*\)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 10
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.....\(.*\)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 9
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '....\(.*\)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 8
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '... \(.*\)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi
if test "$LENGTH" -eq 7
then
    P_1=`expr "$P" : '\(.*\).....'`
    P_2=`expr "$P" : '.. \(.*\)...'`
    P_3=`expr "$P" : '.....\(.*)'`
fi

echo "
                                " $P_1 $P_2 $P_3
echo ""
echo " EDITION DES NUMEROS DE TABLE"
echo "
"
echo " TRAITEMENT ?   [ <— ]=OK, [Del]=Abandon
"
echo "
"
echo ""

cd ../shell_1

if test -f ${LOGNAME}.prt
then rm ${LOGNAME}.prt
fi

SMSO=`tput smso`
RMSO=`tput rmso`
```

```

if [ "$LENGTH" = 0 ]
then

    tput cup 0 0
    echo " CONCOURS : "
    tput cup 0 12
    echo "$SMSO" "$RMSO"
    tput cup 0 12
    read P_1
    if [ "$P_1" = "" ]
    then P_1=0
    fi

    tput cup 0 0
    echo " " "$P_1 $P_2 $P_3"
    tput cup 0 0
    echo " TYPE : "
    tput cup 0 12
    echo "$SMSO" "$RMSO"
    tput cup 0 12
    read P_2
    if [ "$P_2" = "" ]
    then P_2=0
    fi

    tput cup 0 0
    echo " " "$P_1 $P_2 $P_3"
    tput cup 0 0
    echo " SESSION : "
    tput cup 0 12
    echo "$SMSO" "$RMSO"
    tput cup 0 12
    read P_3
    if [ "$P_3" = "" ]
    then P_3=0
    fi
    echo ""

    tput cup 0 0
    echo " " "$P_1 $P_2 $P_3"
fi

# TRAITEMENT ?
tput cup 4 14
echo "$SMSO" "$RMSO"
tput cup 4 14
read REPONSE
tput cup 4 14
echo " "

tput cup 6 0
echo " Traitement en cours... \c"

sacego -q ../ace_1/post_itALL $P_1 $P_2 $P_3 $LOGNAME

cat ${LOGNAME}.prt | sed -f ../bull/isoTOascii.sed > ${LOGNAME}.tmp
mv ${LOGNAME}.tmp ${LOGNAME}.prt
lp ${LOGNAME}.prt > /dev/null

echo "TERMINE !"

```

3.5. Exemple de numéros de table

Cet exemple d'état, reconstitué sous Word à partir de la simulation, propose 22 numéros de table.

<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">1</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. AZNAVOUR Charles</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">7</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. BRUEL Patrick</p>
<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">13</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME TUCK Hailey</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">18</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. LE FORESTIER Maxime</p>

<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">2</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME BAEZ Joan</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">8</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. CABREL Francis</p>
<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">14</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. GOLMAN Jean-Jacques</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center;">19</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME PARADIS Vanessa</p>

<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">3</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. BASHUNG Alain</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">9</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. CLERC Julien</p>
<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">15</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. HALLIDAY Johnny</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">20</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. REGGIANI Serge</p>

<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">4</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME BIRKIN Jane</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">10</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. DELPECH Michel</p>
<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">16</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME JONASZ Michel</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">21</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. SELLER William</p>

<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">6</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. BREL Jacques</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">12</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME FARMER Mylène</p>
<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 1.5em;">PLACE LIBRE</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 1.5em;">PLACE LIBRE</p>

<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">5</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. BRASSENS Georges</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">11</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. DUTRONC Jacques</p>
<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">17</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>M. LAVILLIERS Bernard</p>	<p>CONCOURS INTERNE Salle : 1</p> <p style="text-align: center;">SECRÉTAIRE D'ADMINISTRATION SCOLAIRE ET UNIVERSITAIRE</p> <p style="text-align: center; font-size: 2em;">22</p> <p style="text-align: center;">Numéro de place à reporter sur votre copie</p> <p>MME VARTAN Sylvie</p>

4. Méthodologie de programmation LCP

En savoir davantage sur LCP

Vous trouverez trois billets dans mon blog dont le billet **Méthodologie de programmation LCP** qui m'a permis d'alimenter cette discussion. Ce billet est une ébauche de tutoriel LCP en cours de rédaction.

Adeptes LCP, j'expose dans ce billet mon expérience de la méthode en respectant au mieux la pédagogie originelle. Certains aspects de cette pédagogie qui se réfèrent à des supports de cours des années 70 peuvent décourager le lecteur. Il suffit de les survoler ou de passer outre en utilisant la navigation internet « *Rechercher dans la page* ».

Les 3 billets consacrés à LCP :

-  [Méthodologie de programmation LCP](#)
-  [La programmation rationnelle](#)
-  [Logique de construction des programmes](#)

Et pour en savoir encore plus :

-  [La logique informatique](#) (Blog de Serge Meneut)

FIN – Devoir de vacances d'algorithmique (Force 6/7)
Imprimer des numéros de table en piles

Club des professionnels en informatique

Le plus important club en langue française pour les professionnels en informatique

Le club **Developpez.com** est le principal site d'entraide des informaticiens francophones avec jusqu'à 4 millions de visiteurs uniques par mois. Le club est fréquenté par des développeurs, chefs de projets, DBA, responsables IT, etc.

Le Club vous apporte gratuitement une multitude d'avantages :

[réseau social](#), [newsletter](#), [magazine](#), [actualités](#), [plus de 14 000 cours](#), [tutoriels](#), [articles](#), [plus de 100 FAQ's](#), [tests](#), [comparatifs](#), [débats](#), [sondages](#), [outils](#), [sources](#), [composants et exemples de codes](#), les [BLOGs](#), et enfin la [TV](#). Le Club est animé bénévolement par tous les membres du Club et en particulier par [l'équipe de rédaction du Club](#). L'inscription est facile, rapide et gratuite : [Inscrivez-vous](#).

À qui s'adresse le Club ?

Le Club a pour vocation de vous aider, quels que soient votre niveau et les technologies, langages ou outils que vous utilisez. Cependant, le Club se focalise en priorité sur la [formation](#) et l'aide aux professionnels en informatique, ou ceux qui souhaitent le devenir, comme les étudiants par exemple. Si vous êtes développeur amateur ou occasionnel, vous êtes aussi bienvenu au Club, si comme tous les membres du Club, vous acceptez de suivre les [règles du Club](#) qui ont pour but de privilégier l'efficacité et la bonne ambiance : courtoisie, lisibilité, précision, etc.

Le plus important forum pour les professionnels en informatique en langue française

Le [forum](#), avec plus de 5 000 messages quotidiens, est le principal lieu d'entraide des professionnels en informatique francophones. Le forum est largement subdivisé pour répondre au mieux aux activités des différentes technologies, et répondre à toutes vos attentes. Le forum est modéré par une équipe de bénévoles dans le but de vous faciliter la lecture du forum et vous offrir le maximum d'efficacité : suppression des spams et autres pollutions, déplacement des messages hors sujet, édition ou suppression des messages indécents ou à caractères illégaux, etc. C'est donc non seulement le forum en langue française le plus actif pour les professionnels en informatique, mais c'est aussi le plus rapide, le plus courtois, le plus efficace, et aussi le plus riche en fonctionnalités.

