EMD1 d'ALGORITHMIQUE

Date : samedi 11 décembre 2010
Durée : 2 Heures
DOCUMENTS INTERDITS

EXERCICE 1 (15 points) :

Il existe des nombres mystérieux qui, lorsque vous les inversez et vous les élevez au carré, eux et leurs inverses, vous obtenez deux nombres composés par les mêmes chiffres. Regardez l'exemple, c'est plus facile à comprendre !

Exemples :

- si on prend le nombre 12, son inverse est le nombre 21. Si on élève 12 au carré, on obtient $12^2 = 144$, et si on élève aussi au carré son inverse $21^2 = 441$. On observe que 144 et 441 sont composés des mêmes chiffres.
- Si on prend le nombre 102, on s'aperçoit que $102^2 = 10404$ et $201^2 = 40401$

Pouvez-vous retrouver tous les nombres inférieurs à un nombre donné qui satisfont cette bizarrerie ?

EXERCICE 2 (5 points) :

- Nous sommes des nombres de 5 chiffres
- Notre chiffre des dizaines est égal à celui des centaines
- Notre chiffre des milliers est égal à deux fois celui des dizaines
- Notre chiffre des centaines est égal à la moitié de celui des dizaines de milliers
- La somme des chiffres de chacun de nous est un nombre premier supérieur à 30

Combien et qui sommes nous ?
Nous rechercher à la main est un vrai casse tête chinois, alors sauriez-vous construire **l'analyse seulement de l' algorithme principal** qui permet de nous retrouver facilement ?

Travail à Faire :

- EXERCICE 2 : à traiter complètement
- EXERCICE 1 : Ne donner que l'analyse de l'algorithme principal
- Un bonus de 3 points vous sera accordé si vous programmez l'algorithme principal de l'exercice 2 (1/2 point sera ôté par erreur)

ATTENTION: Votre solution doit **ABSOLUMENT** et tenir compte du **formalisme étudié** en cours et vous pouvez utiliser ou non la modularité, bien que nous vous la conseillons !
De plus, deux **(2) points seront déduits de la note pour toute copie non soignée.**
Alors soignez votre travail et bon courage !

ESI CPI 1 - Corrigé EMD 1 du 11 décembre 2010

BAREME

TOUT COPIE NON SOIGNEE DOIT SERA SANCTIONNEE (max. – 2 points)

EXO_1 (10 points)

- Boucle générale (1 pt)
- Inversion (3pts)
- Meme_chiffres (6 pts)
  - Extraction des positions (1pt)
  - Calcul de la fréqence (2pts)
  - Cohérence de l'ensemble (3pts)

EXO_2 (10 points)

	ANALYSE	ALGORITHME
Boucle générale + comptage des Nbs + écritures	1	1
Extractions des positions nécessaires	2	2
Somme des positions d'un nbre	1/2	1/2
Verif. Nbre premier	1/2	1/2
Conformité avec l'analyse, respect du formalisme, soin		2
	4 pts	6pts

PROGRAMMATION : dans le contexte et à corriger seulement si l'analyse et les algorithmes sont terminés.
(-0.5 pt./erreur)
Il s'agit d'un BONUS, il faut donc qu'il soit mérité.

ANALYSE GENERALE DE EXERCICE 1

Nota : que l'on utilise la modularité ou pas cette analyse générale reste valable.

- Donner Nb
- Nous allons prendre tous les nombres A compris entre 1 et N , et pour chacun d'eux :
  - On **inverse** A pour obtenir le nombre B
  - On élève A au carré
  - On élève B au carré
  - Si A² et B² sont **composés des mêmes chiffres**, on écrit A et B .

ANALYSE DETAILLEE DE L' EXERCICE 1 SANS LA MODULARITE

- Donner N
- Nous allons prendre tous les nombres A compris entre 1 et N , et pour chacun d'eux :

Inversion de A pour obtenir le nombre B (*Voir exemple)*

- On divise successivement A par 10, et on s'arrête quand le quotient est égal à 0, le nombre de divisions nous donne le nombre de positions (Npos) ***(on calcule le nombre de positions de A)***
- X = (10*10*10*………10)(Npos – 1) fois ***(Calcul de x = 10^{npos-1})***
- On divise successivement A par 10, jusqu'à obtenir un quotient = 0 et à chaque fois, on calcule :
  B = (B A mod 10)*X
  X = X div 10 ***(on inverse A –)***
- On élève A au carré (A2)
- On élève B au carré (B2)

Vérification si A et B sont composes des mêmes chiffres

- On divise successivement A2 par 10, et on s'arrête quand le quotient est égal à 0, le nombre de divisions nous donne le nombre de positions (NposA2) ***(on calcule le nombre de positions de A2)***
- On divise successivement B2 par 10, et on s'arrête quand le quotient est égal à 0, le nombre de divisions nous donne le nombre de positions (NposB2) ***(on calcule le nombre de positions de B2)***
- Si nposa2 <> nposb2 on fait aig= FAUX ***(aig est un aiguillage qui sera égal à VRAI lorsque A2 et B2 sont composés des mêmes chiffres)***
- Si nposa2 = nposb2 ***(A2 et B2 contiennent le même nombre de chiffres)***
  - ✓ On répète
    - i:=1;
    - on extrait la position(Posi) I de A2
    - on divise successivement A2 par 10, jusqu'à quotient =0
      - si A2 mod i = posi ***on incrémente le compteur Fa (qui va contenir la fréquence d'apparitions du chiffre Posi dans A2)***
    - on divise successivement B2 par 10, jusqu'à quotient =0
      - si B2 mod i = posi ***on incrémente le compteur Fb (qui va contenir la fréquence d'apparitions du chiffre Posi dans B2)***
    - on incrémente i ***(pour extraire la position suivante)***
    - a2:=a2 div 10 ***(prochaine valeur de A2)***
    - Si fa = fb ,aig = VRAI ***(les fréquences sont identiques)***
    - Si fa <> fb on fait aig = FAUX ***(les fréquences ne sont pas identiques)***

    Jusqu'à ce que (aig = FAUX) OU (i> nposa2)

```
o  Si  (aig = VRAI) on écrit A et B
```

N= 5961 | 10

$1000=10^{npos-1}$ * ……….. 1 | 596 | 100

$+\ 600=10^{npos-2}$ * ……………. 6 | 59 | 10

$+\ 90=10^{npos-3}$ * ……………………. 9 | 5 | 10

$+\ 5=10^{npos-4}$ * ………………………….. 5 | 0 → condition d'arrêt

= 1695

Exemple d'inversion d'un nombre

ANALYSE DETAILLEE DE L' EXERCICE 1 AVEC LA MODULARITE

DECOUPAGE MODULAIRE

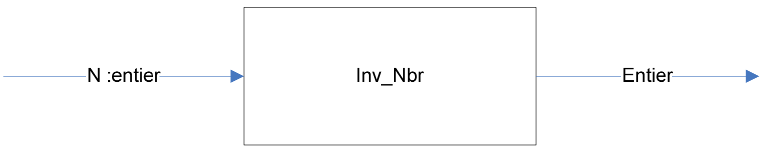
On aura besoin d'un module qui :

- un nombre qui inverse un nombre (Module Inv_Nbr)
- nous permet de savoir si 2 nombres sont composés des mêmes chiffres (Module MEME_CH)

On aura aussi besoin de:

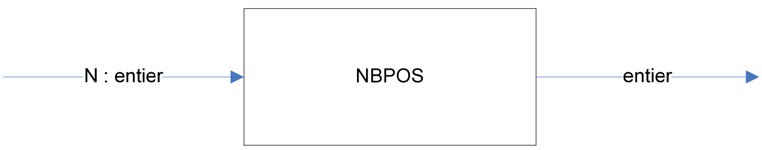
- connaître le nombre de positions d'un nombre (Module Nb_pos)
- Extraire les chiffres d'un nombre (Module Extr_Nb)
- Concaténer deux nombres (Module concat)
- Calculer la fréquence d'apparitions d'un chiffre dans un nombre (module Freq_ch)

FONCTION



Rôle : Inverse le nombre N

FONCTION



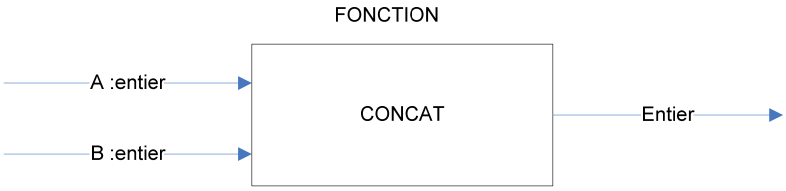
Rôle : Donne le nombre de chiffres composant N

FONCTION

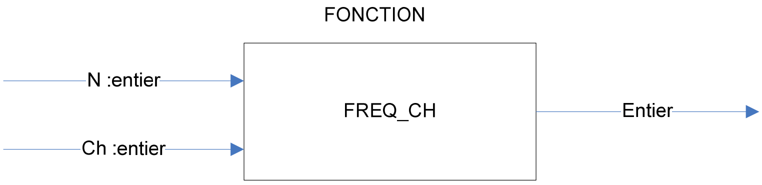
Rôle : Extrait de N, un nombre de L positions à partir de la position P

FONCTION

Rôle : Donne vrai si A et B sont composés des mêmes chiffres



Rôle : Concatène les nombres A et B



Rôle : Donne la fréquence d'apparitions du chiffre Ch dans le nombre N

ANALYSE EXERCICE 1

- Donner Nb
- Nous allons prendre tous les nombres A compris entre 1 et N , et pour chacun d'eux :
  - B = Inv_Nb (A) ***(On inverse A pour obtenir le nombre B)***
  - On élève A au carré
  - On élève B au carré
  - Si MEME_CH (A^2 , B^2) = Vrai on écrit A et B ***(Si A^2 et B^2 ont composés des mêmes chiffres, on les écrit)***

OBSERVATION *: Les solutions ont été données sans et avec la modularité. Vous constatez ainsi que la modularité apporte plus de facilité dans nos solutions en plus de la réutilisation des modules (et donc du code) déjà construits. On se rend compte aisément dans la solution de ces exercices qu'il est plus facile de construire des petits modules qui ne font qu'un petit traitement plutôt qu'une solution unique ou tous les traitements sont enchevêtrés, et dans laquelle si l'on fait une erreur toute notre solution sera fausse alors qu'en découpant le problème en plusieurs parties (modules) nous pouvons en avoir certaines qui sont justes et on fixera notre attention sur les parties fausses seulement.*

EXERCICE 2

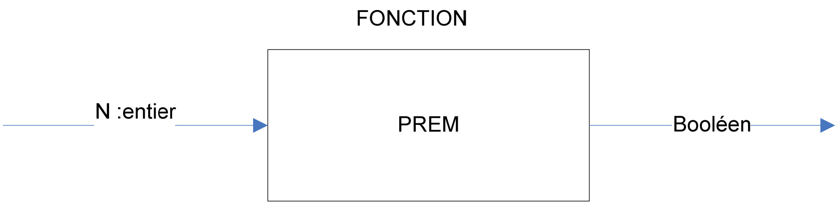
ANALYSE : Nous présentons la solution avec la modularité. La solution sans la modularité est identique, sauf qu'il faudra intégrer dans l'analyse :

1. l'extraction des différentes positions nécessaires
2. comment faire la somme des positions d'un nombre
3. et comment vérifier qu'un nombre est premier ou pas

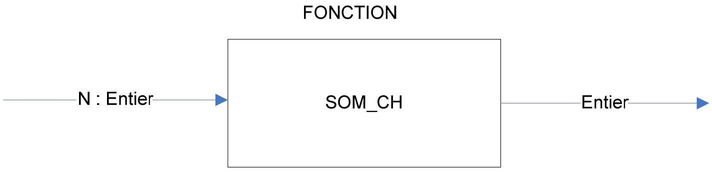
Découpage**:**

On aura besoin :

- **de savoir si un nombre est premier - module PREM**
- **d'extraire les chiffres d'un nombre – module EXTPOS (voir exo précédent)**
- **de faire la somme des chiffres d'un nombre - module SOM_CH**



Rôle : Donne Vrai si N est une nombre premier



Rôle : Donne la somme des chiffres de N

ANALYSE DE L'ALGORITHME PRINCIPAL

- On fait varier i de 100 000 à 99 999 (*pour prendre les nombres de 5 chiffres)*
    - `On extrait le chiffre des dizaines` ***(diz:=extpos(i,1,4))***
    - `On extrait le chiffre des centaines` (**cent:=extpos(i,1,3))**
    - `On extrait le chiffre des muilliers` ***(mille:=extpos(i,1,2))***
    - `On extrait le chiffre des dix milles` ***(dixmille:=extpos(i,1,1))***
    - `Si (diz=cent) ET (mille =2 * diz) ET(cent =dixmille div 2) ET (prem(som_ch(i))) , on écrit i`

```
ALGORITHME  ed1B1011
variables diz, cent, mille, dixmille,I, cpt : Entier
Fonctions extpos, prem , som_ch

DEBUT
Cpt ⇦ 0
Pour  i Allant de  10000 à  99999 Faire
        DPOUR
        Diz ⇦ extpos(i,1,4)
        Cent ⇦ extpos(i,1,3)
        mille ⇦ extpos(i,1,2)
        dixmille ⇦ extpos(i,1,1)
        Si (diz=cent) ET  (mille=2*diz)ET
           (cent =dixmille div 2) ET (prem(som_ch(i)))Alors
                             Dsi
                             Ecrire(i)
                             Cpt ⇦ Cpt +1
                             Fsi
        FPOUR
Ecrire (Cpt)
FIN
```

```
program ed1B1011;
var diz, cent, mille, dixmille,i,cpt:longint;
{$i E:\algo\modules\extpos.fon}
{$i E:\algo\modules\prem.fon}
{$i E:\algo\modules\som_ch.fon}
BEGIN
Cpt:=0;
for i := 10000 to 99999 do
        BEGIN
        diz:=extpos(i,1,4);
        cent:=extpos(i,1,3);
        mille:=extpos(i,1,2);
        dixmille:=extpos(i,1,1);
        if (diz=cent) and (mille=2*diz)and (cent =dixmille div 2) and
           (prem(som_ch(i)))then
                                 bEGIN
                                 writeln(i);
                                 Cpt := cpt +1 ;
                                 end.
        END;
Write('le nombre de nombres est :', cpt);
readln ;
END.
```