

Lycée secondaire Rue de Russie	<i>Devoir de contrôle</i> <i>N° 5</i>	Année scolaire : 2006 / 2007
Professeur : Mohamed TRABELSI		Durée : 1h - Théorie
Matière : Algorithmique		Classe : 3 ^{ème} SI

Nom & prénom :

Note : /20

Énoncé : Algorithmes arithmétiques

Deux entiers strictement positifs m et n sont dits nombres amis si et seulement si :

- La somme des diviseurs de m sauf lui-même est égale à n
- Et La somme des diviseurs de n sauf lui-même est égale à m.

Faire l'analyse d'un programme intitulé **nombres_amis** permettant de saisir deux entiers strictement positifs m et n et d'afficher un message indiquant s'il s'agit de deux nombres amis ou non.

Exemple :

220 et 284 sont deux nombres amis. On faisant le calcul on a :

- La somme des diviseurs de 220 sauf lui même = {1, 2, 4, 5, 10, 11, 20, 22, 44, 55, 110} = 284
- La somme des diviseurs de 284 sauf lui même = {1, 2, 4, 71, 142} = 220

T.A.F :

1. Analyse principale.
2. Créer les modules suivants dans votre analyse :
 - a. Module **saisie** **()**, pour saisir m et n
 - b. Module **somme_diviseurs** **()**, ce module reçoit comme paramètre un nombre entier > 0 et retourne la somme de ses diviseurs.
 - c. Les traitements restant sont à faire dans l'analyse principale.
3. En déduire les algorithmes au fur et à mesure.

Lycée secondaire Rue de Russie	<i>Correction</i> <i>Devoir de contrôle</i> <i>N° 5</i>	Année scolaire : 2006 / 2007
Professeur : Mohamed TRABELSI		Durée : 1h - Théorie
Matière : Algorithmique		Classe : 3 ^{ème} SI

1. Analyse principale :

Résultat : Affichage message

Traitements :

Si somme des diviseurs de $n = m$ ET somme des diviseurs de $m = n$ Alors m et n sont des nombres amis sinon ils ne le sont pas.

Faire appel à la fonction `somme_diviseurs (x)` pour le calcul de cette somme.

Saisie (n, m)

Algorithme

0) Début **Nombres_amis**

1) saisie (n, m)

2) **Si** (`somme_diviseurs (n) = m`) Et (`somme_diviseurs (m) = n`)

Alors écrire ($n, "$ et ", $m, "$ Sont des nombres amis")

Sinon écrire ($n, "$ et ", $m, "$ ne sont pas des nombres amis")

Fin si

3) Fin **Nombres_amis**

T.D.O Globaux

objet	Type	Rôle
m	Mot	
n	mot	

2. Analyse de la fonction `somme_diviseurs` :

Fonction `somme_diviseurs (x : mot) : mot`

Résultat : `somme_diviseurs`

Traitement :

`somme_diviseurs` ← `sdx`

On initialise `sdx` par 1, (1 étant toujours un diviseur)

Puis grâce à une boucle "pour" allant de 2 à $x-1$ on calcul le cumul des diviseurs de x .

Algorithme

- 0) Fonction **somme_diviseurs** (x : mot) : mot
- 1) sdx ← 1
 Pour i de 2 à x-1 faire
 Si x MOD i = 0 alors sdx ← sdx + i
 Fin si
 Fin pour
- 2) somme_diviseurs ← sdx
- 3) Fin **somme_diviseurs**

T.D.O Locaux

Objet	Type	Rôle
sdx	mot	Somme des diviseurs de x
i	mot	Compteur

3. analyse de la procédure saisie :

Procédure **saisie** (var n, m : mot)

Résultat : n et m saisis

Traitement :

 Répéter
 N, m = donnée
 Jusqu'à (n>0) et (m>0)

Algorithme

- 0) Procédure **saisie** (var n, m : mot)
- 1) Répéter
 Ecrire ("Donner n et m : "), lire (n, m)
 Jusqu'à (n>0) et (m>0)
- 2) Fin **saisie**

4. Traduction Pascal :

{DC n°5 3 SI 06-07, Algorithmes Arithmétiques, Théorie}

```
program nombres_amis;
uses wincrt;

var
  n,m:word;

procedure saisie(var m,n:word);
begin
  repeat
    write('Donner deux entier strictement positifs : ');
    readln(n,m);
  until (n>0) and (m>0);
end;

function somme_diviseurs(x:word):word;
var
  sdx:word;
  i :word;
begin
  sdx:=1;
  for i:=2 to x-1 do
    if x mod i=0 then sdx:=sdx+i;
  somme_diviseurs:=sdx;
end;

BEGIN
  Saisie (n,m);

  if (somme_diviseurs(n) = m) and (somme_diviseurs(m) = n)
    Then write (n,' et ',m,' sont des nombres amis')
    else write (n,' et ',m,' ne sont pas des nombres amis');
end.
```