

Classe : 4^{ème} SI 1

Durée : 1h 30mn

Prof : Mr.Karchoud Radhouane
2024

Nb pages : 3 Coeff. : 03

Date : Février

Nom et prénom :**Exercice n°1: (6 pts)**

Un facteur dominant dans une décomposition en facteurs premiers, est le facteur le plus utilisé dans la décomposition. En cas d'égalité, on suppose que le premier facteur est le dominant.

Exemple :

N = 7452, sa décomposition en facteurs premiers est $2 \times 2 \times 3 \times 3 \times 3 \times 3 \times 23$, donc le facteur dominant est 3.

N = 188640, sa décomposition en facteurs premiers est $2 \times 2 \times 2 \times 2 \times 2 \times 3 \times 3 \times 5 \times 13$, donc le facteur dominant est 2.

N = 9523, sa décomposition en facteurs premiers est 89×107 , donc le facteur dominant est 89.

On dispose d'un fichier texte nommé "**nombres.txt**" remplis par N nombres à raison d'un nombre par ligne.

Travail demandé :

- 1- Ecrire l'algorithme d'un module qui permet de former un fichier binaire "**dominant.dat**" par des enregistrements.

Chaque enregistrement est formé par deux champs :

Nbr : entier à prendre du premier fichier.

Fact : le facteur dominant du champ Nbr.

- 2- Ecrire l'algorithme d'un module qui permet d'afficher le facteur le plus dominant dans le fichier "**dominant.dat**".

Exercice n°2: (10 pts)

- 1- Soit la fonction récursive suivante :

Fonction F(n,d : entier) :

Début

Si d = 0 alors

Retourner 0

Sinon si (n MOD d = 0) alors

Retourner d + F(n,d-1)

Sinon

Retourner F(n,d-1)

Fin si

Fin

Questions :

- a) Compléter le type de la fonction (réécrire l'entête sur la copie).
- b) Sachant que l'appel de la fonction est de la forme suivante : $x \square F(n, n \text{ DIV } 2)$
Exécuter manuellement (donner la trace d'exécution) la fonction F pour n = 9 et n = 5.
- c) Déduire le rôle de la fonction F.

2-

Si vous prenez un nombre entier naturel $N > 1$ et vous calculez la somme de ses diviseurs propres (c'est-à-dire excepté lui-même), et vous refaites la même opération sur ce résultat, vous obtenez une **suite aliquote**.

Exemple : si N = 24 La suite aliquote obtenue est : N = 24, 36, 55, 17, 1

Généralement une suite aliquote s'arrête lorsque l'on arrive à 1 (car 1 n'a pas de diviseur propre). Lorsque le premier élément calculé est égal à 1, N est un nombre premier. Mais ce n'est pas toujours le cas, il arrive que la suite aliquote soit fermée et compte e éléments calculés, dans ce cas, N est ce que l'on appelle un

nombre sociable d'ordre e . De plus si l'ordre e est égal à 1, N est un nombre parfait et si e est égal à 2, N est un nombre amical.

Définitions :

- Une suite aliquote fermée est une suite dont le dernier élément est égal au premier élément donné de la suite. Son ordre e est égal au nombre d'éléments calculés de la suite (c.à.d. excepté le premier) (*exemples 3, 4 et 5*)
- Un nombre parfait est un nombre dont la somme des diviseurs propres est égale au nombre lui-même. (*exemple 3*)
- Deux nombres A et B sont dits amicaux si la somme des diviseurs de A est égale à B et la somme des diviseurs de B est égale à A . *exemple 4*)
- Un nombre premier est un nombre qui n'accepte comme diviseurs que 1 et lui-même (*exemple 2*).

Exemple 1 :

```
donnez N :24
Suite aliquote : 24,36,55,17,1,
```

```
donnez N :11
Suite aliquote : 11,1,*N est un nombre premier *
```

```
donnez N :28
Suite aliquote : 28,28,*N est sociable et parfait d'ordre 1 *
```

```
donnez N :220
Suite aliquote : 220,284,220,*N est sociable et amical d'ordre 2 *
```

```
donnez N :12496
Suite aliquote : 12496,14288,15472,14536,14264,12496,*N est sociable d'ordre 5 *
```

aliquote pour un entier N donné.

- b) On dispose d'un fichier binaire "**nombres.dat**" formé par des enregistrements de la forme suivante :
- o Val est un entier > 1 déjà saisi.
 - o Premier : champ booléen qui indique si le champ Val est un entier premier ou non.
 - o Parfait : champ booléen qui indique si le champ Val est un entier parfait ou non.
 - o Amical : champ booléen qui indique si le champ Val est un entier amical ou non.

L'état initial de ce fichier est de la forme suivante :

24	17	6	220	12496	28

- ✓ On vous donne l'algorithme du programme principal suivant :

Algorithme Exercice2

Début

Np □ "nombres.dat"

Ouvrir(Np, fb, "wb")

Remplir (fb) # **NB** : cette procédure permet de remplir uniquement le champ Val.

Compléter (fb) # **NB** : cette procédure permet de compléter les autres champs du fichier.

..... # **NB** : appel d'une procédure qui permet d'afficher le fichier.

Fin

- ✓ Compléter le TDNT qui correspond à l'exercice.

Type
Aliquote = enregistrement
.....
.....

.....
.....
.....
.....
.....

✓ Développer le module **Completter (fb)** qui permet de remplir les champs Premier, Parfait et Amical.
L'état final de ce fichier est de la forme suivante :

24	17	6	220	12496	28
Faux	Vrai	Faux	Faux	Faux	Faux
Faux	Faux	Vrai	Faux	Faux	Vrai
Faux	Faux	Faux	Vrai	Faux	Faux

✓ Développer le module d'affichage du fichier "nombres.dat".

Exercice n°3: (4 pts)

Divisibilité par 21

Le nombre $a_n \dots a_1 a_0$ est divisible par 21 si et seulement si $a_n \dots a_1 - 2a_0$ (ou sa valeur absolue) l'est. Pour voir si un nombre est divisible par 21, il suffit de la répéter cette transformation jusqu'à obtenir un résultat strictement inférieur à 21. Le nombre de départ est divisible par 21 si et seulement si le résultat final est 0.

Exemple

Le nombre 5 289 417 est divisible par 21 car
 $528\ 941 - 2 \times 7 = 528\ 927$,
 $52\ 892 - 2 \times 7 = 52\ 878$,
 $5\ 287 - 2 \times 8 = 5\ 271$,
 $527 - 2 \times 1 = 525$,
 $52 - 2 \times 5 = 42$ et
 $4 - 2 \times 2 = 0$.

Travail demandé :

Pour vérifier si un entier $N > 10$ est divisible par 21 ou non, on doit suivre les étapes suivantes :

- 1- Remplir la première ligne d'une matrice M par les chiffres de N (chaque chiffre dans une case).
- 2- Ensuite, remplir les autres lignes par les résultats des calculs comme indiqué au-dessus (chaque chiffre dans une case)
- 3- La dernière ligne indique la divisibilité.

Exemple :

Introduire un entier à vérifier: 5289417
5 2 8 9 4 1 7
un entier à vérifier: 745

pas divisible par 21

5289417 est divisible par 21

Ecrire un algorithme d'un module qui permet de vérifier si un entier N donné en paramètre est divisible par 21 ou non ?

BON TRAVAIL