

Exercice : On se propose d'écrire l'algorithme du problème qui permet de :

- Saisir un entier naturel strictement positif N1
- Saisir un entier naturel strictement positif N2
- Vérifier et afficher si N1 et N2 sont cousins. Deux entiers positifs N1 et N2 sont cousins si le produit des chiffres non nuls de N1 est égal au produit des chiffres non nuls de N2

Exemples :

| Données : | Résultat |
|------------------------|---|
| 1) N1=1023 et N2=10231 | 1023 et 10231 sont cousins (car $1*2*3=1*2*3*1$) |
| 2) N1=4020 et N2=10221 | 4020 et 10221 ne sont pas cousins (car $4*2 \neq 1*2*2*1$) |

Procédure *Saisie_n(@ n : entier)*

Début

Répéter

Ecrire("Donner n > 0 ? ")

Lire(n)

Jusqu'à (n > 0)

| TDOL : | |
|--------|-------------|
| Objets | Type/Nature |
| | |
| | |

Fin

#.....#

Fonction *Calcul_p(n : entier) : entier*

Début

 ch \leftarrow **ConvCh**(n)

 p \leftarrow 1

Pour i **de** 0 **à** long(ch)-1 **faire**

 v \leftarrow **Valeur**(ch[i])

Si v \neq 0 **alors**

 p \leftarrow p * v

Fin Si

Fin Pour

Retourner p

| TDOL : | |
|---------|----------------------|
| Objets | Type/Nature |
| ch | Chaine de caractères |
| p, i, v | entiers |

Fin

#.....#

Procédure *Affiche(a : entier, b : entier)*

Début

Si Calcul_p(a) = Calcul_p(b) **alors**

Ecrire(a, "et", b, "sont cousins")

Sinon

Ecrire(a, "et", b, "ne sont pas cousins")

Fin Si

Fin

| TDOL : | |
|----------|-------------|
| Objets | Type/Nature |
| Calcul_p | Fonction |

Algorithm Test_Cousins

Début (programme principal)

Saisie_n(x)

Saisie_n(y)

Affiche(x,y)

Fin

TDOG :

| Objets | Type/Nature |
|----------|-------------|
| x, y | Entiers |
| Saisie_n | Procédure |
| Affiche | Procédure |