

Février 2022

Durée : 1h30

EMD 1 ALGORITHMIQUE

Exercice 1 (08 points)

Un nombre **N** est dit « **nombre Vampire** » s'il est égale à un produit de ses chiffres. Exemple : **N = 126** est un nombre Vampire, car **126 = 21*6**.

On peut vérifier si un nombre **N** à trois chiffres est Vampire en extrayant les trois chiffres qui le composent. Si tous ces derniers sont différents de zéro, alors il suffit de s'assurer qu'un de ces chiffres divise **N** et que le nombre résultant de la division est composé des deux chiffres restants. Si un tel cas se présente, alors **N** est Vampire.

1. Écrire un algorithme qui affiche tous les nombres Vampires positifs à trois chiffres.
2. Écrire le programme Pascal correspondant à votre réponse à la question 1.

Exercice 2 (12 points)

Soit un vecteur **V** de **n** éléments de type entier :

1. Écrire une procédure **remplirVecteur** qui permet de remplir **V**.
2. Écrire une fonction **compterOccurence** qui permet de compter le nombre d'apparitions d'une valeur donnée dans **V**.
3. Écrire une procédure **remplacerOccurence** qui modifie **V** de sorte que pour chaque valeur distincte de **V**, seule la première occurrence trouvée sera gardée et toutes les autres occurrences seront remplacées par la valeur zéro (0).

Illustration : Pour le vecteur **V** de **n = 10** éléments :

3	5	5	3	3	3	2	1	4	5
---	---	---	---	---	---	---	---	---	---

on obtient après le traitement demandé le vecteur :

3	5	0	0	0	0	2	1	4	0
---	---	---	---	---	---	---	---	---	---

4. Écrire une procédure **deplacerZeros** qui déplace toutes les valeurs nulles de **V** à droite du vecteur.

Illustration : Pour le vecteur **V** de **n = 10** éléments :

6	0	4	4	0	0	1	8	4	0
---	---	---	---	---	---	---	---	---	---

on obtient après le traitement demandé le vecteur :

6	4	4	1	8	4	0	0	0	0
---	---	---	---	---	---	---	---	---	---

5. Écrire une procédure **trierDecroissant** qui trie les valeurs de **V** selon l'ordre décroissant.
6. Écrire un **algorithme** qui lit un vecteur, et qui trie et affiche les valeurs distinctes du vecteur selon l'ordre décroissant.

Exemple : Pour le vecteur de **10** éléments :

8	8	4	4	6	6	4	1	8	2
---	---	---	---	---	---	---	---	---	---

l'algorithme doit afficher la suite de valeurs : **8 6 4 2 1**.

Corrigé de l'EMD 1 ALGORITHMIQUE

Exercice 1 : (noté sur 8 pts avec : 6 pts pour la question 1 et 2 pts pour la question 2)

1. Ecriture de l'algorithme qui affiche tous les nombres Vampires positifs à trois chiffres.

Algorithme exercicel;

Var

m, r1,r2,r3,x,y : entier;
vampire : Booléen;

Début

Pour m allant de 100 jusqu'à 999 faire (1 pt)

r1 ← m Mod 10; (0,25 pt)

r2 ← (m Div 10) Mod 10; (0,25 pt)

r3 ← m Div 100; (0,25 pt)

vampire ← faux; (0,25 pt)

Si ((r1 > 0) et (r2 > 0) et (r3 > 0)) **alors** (0,5 pt)

Si (m Mod r1 = 0) **alors** (1 pt)

x ← (m Div r1) Div 10 ;

y ← (m Div r1) Mod 10 ;

Si (x=r3) et (y=r2) ou (x=r2) et (y=r3) **alors**

vampire ← vrai;

Sinon

Si (m Mod r2 = 0) **alors** (1 pt)

x ← (m Div r2) Div 10 ;

y ← (m Div r2) Mod 10 ;

Si (x=r1) et (y=r3) ou (x=r3) et (y=r1) **alors**

vampire ← vrai;

Sinon

Si (m Mod r3 = 0) **alors** (1 pt)

x ← (m Div r3) Div 10 ;

y ← (m Div r3) Mod 10 ;

Si (x=r1) et (y=r2) ou (x=r2) et (y=r1) **alors**

vampire ← vrai;

Fin Si;

Fin Si;

Fin Si;

Fin Si;

Si (vampire = vrai) **alors** (0,5 pt)

Ecrire (m, ' est un nombre Vampire');

Fin Si ;

Fin Pour ;

Fin ;

2. Traduction de l'algorithme en un programme Pascal. (2 pts)

Program Exercicel;

Uses WinCRT ;

Var

m, r1,r2,r3,x,y : integer;
vampire : Boolean;

Begin

For m:=100 **To** 999 **Do**

Begin

r1 := m Mod 10;

r2 := (m Div 10) Mod 10;

r3 := m Div 100;

vampire := false;

If ((r1 > 0) And (r2 > 0) And (r3 > 0)) **Then**

Begin

If (m Mod r1 = 0) **Then**

Begin

x := (m Div r1) Div 10 ;

y := (m Div r1) Mod 10 ;

If (x=r3) And (y=r2) Or (x=r2) And (y=r3) **Then**

vampire := true;

End

Else

If (m Mod r2 = 0) **Then**

Begin

x := (m Div r2) Div 10 ;

y := (m Div r2) Mod 10 ;

If (x=r1) And (y=r3) Or (x=r3) And (y=r1) **Then**

vampire := true;

End

Else

If (m Mod r3 = 0) **Then**

Begin

x := (m Div r3) Div 10 ;

y := (m Div r3) Mod 10 ;

If (x=r1) And (y=r2) Or (x=r2) And (y=r1) **Then**

vampire := true;

End;

End;

If (vampire = true) **Then**

writeln (m, ' est un nombre Vampire')

End ;

End.

Exercice 2 : (noté sur 12 pts selon le barème détaillé ci-dessous)

// L'étudiant peut écrire les sous-algorithmes séparément, et faire des appels aux sous-algorithmes dans le corps de l'algorithme.

Algorithme exercice2; (0,5 pt pour la forme générale de l'algorithme)

Type

vecteur = Tableau [1..10] de entier; (0,5 pt)

Var

V : vecteur;

taille, i : entier ;

Procédure remplirVecteur (**E/S** V : vecteur ; taille : entier) ; (1 pt)

Var

i : entier ;

Début

Pour i allant de 1 à taille **faire**

Ecrire ('Donnez la valeur ', i, ' du tableau ');

Lire (V[i]);

Fin Pour ;

Ecrire ('Merci le tableau est bien rempli');

Fin ;

Fonction compterOccurrence (V : vecteur ; taille : entier ; valeur : entier) : entier ; (2 pts avec 0,5 pt pour la déclaration et 1,5 pts pour le corps de la fonction)

Var

i, occurrence : entier ;

Début

occurrence \leftarrow 0;

i \leftarrow 1;

Tant que (i \leq taille) **faire**

Si (V[i] = valeur) **alors**

occurrence \leftarrow occurrence + 1;

Fin Si;

i \leftarrow i + 1;

Fin Tant que ;

nombreOccurrence \leftarrow occurrence;

Fin;

Procédure remplacerOccurrence (**E/S** V: vecteur; taille: entier) ; (2 pts avec 0,5 pt pour la déclaration et 1,5 pts pour le corps de la procédure)

Var

i, j : entiere ;

Début

Pour i allant de 1 à taille **faire**

Si (compterOccurrence (V, taille, V[i]) > 1) et (V[i] <> 0) **alors**

Pour j allant de i+1 à taille **faire**

Si (V[j] = V[i]) **alors**

V[j] \leftarrow 0;

Fin Si ;

Fin pour ;

Fin Si ;

Fin Pour;

Fin ;

Procédure deplacerZeros(**E/S** V : vecteur; taille : entier); (2 pts avec 0,5 pt pour la déclaration et 1,5 pts pour le corps de la procédure)

Var

T : vecteur ;

i, j: entiere ;

Début

Pour i allant de 1 à taille faire

T[i] ← 0;

Fin Pour ;

i ← 1;

j ← 1;

Tant que (i <= taille) faire

Si (V[i] <> 0) alors

T[j] ← V[i];

j ← j + 1;

Fin Si ;

i ← i + 1;

Fin Tant que ;

V ← T;

Fin ;

Procédure trierDecroissant (**E/S** V : vecteur ; taille : entier) ; **(2 pts avec 0,5 pt pour la déclaration et 1,5 pts pour le corps de la procédure)**

Var

M, i, j : entier ;

Début

Pour i allant de 1 à taille-1 faire

Pour j allant de i+1 à taille faire

Si (V[i] < V[j]) alors

M ← V[i];

V[i] ← V[j];

V[j] ← M;

Fin Si ;

Fin pour ;

Fin pour ;

Fin ;

Début

n ← 10; **(0,25 pt)**

// La valeur 10 est non exigée, l'étudiant peut donner la valeur qu'il veut pour la variable n, à condition de ne pas causer un débordement du vecteur lors de sa lecture.

remplirVecteur (V, n); **(0,25 pt)**

remplacerOccurrence (V, n); **(0,25 pt)**

deplacerZeros(V, n); **(0,25 pt)**

trierDecroissant (V, n); **(0,25 pt)**

i ← 1;

Tant que (V[i] <> 0) faire **(0,75 pt)**

Ecrire (V[i]);

i ← i + 1;

Fin tant que ;

Fin.