TP N3 EN ALGO

07/12/2023

Exercice 1 :
Écrire un programme qui calcule le produit de tous les éléments d'un tableau d'entiers.

```
Algorithme Produit
Variables :
        produit :  tab[];
        n, p, i : entier;

DÉBUT

        Afficher ("la taille de votre tableau");
        lire (n);
        p = 1;
        Afficher ("veuillez entrer les valeurs de votre tableau");
        POUR i = 1 JUSQU'À n FAIRE
           lire (tab[i]);
           p = p * tab[i];
        FINPOUR
        Afficher (“Le produit des valeurs de votre tableau est : “,p);

FIN
```

Exercice 2 :
Écrire une fonction qui supprime les éléments en double d'un tableau et renvoie le tableau sans doublons.

```
Algorithme suppression
Variables :
        mon_tableau = [1, 2, 7, 3, 4, 9, 17, 100, 100, 100,  4, 5];
        resultat : tab[];

Début

Fonction supprimer_doublons(tableau)
    tableau_sans_doublons = [];
   pour chaque i dans tableau faire
       si i n'est pas dans tableauSansDoublons
          ajouter i à tableauSansDoublons;
       finsi
  finpour
  retourner tableau_sans_doublons;

Fin Fonction
```

```
resultat = supprimer_doublons(mon_tableau);
Afficher (“Tableau sans doublons : ”)
POUR i = 1 JUSQU'À taille(mon_tableau) FAIRE
   Afficher (mon_tableau);
FINPOUR
Fin
```

Exercice 3 :
Implémenter un algorithme pour faire une rotation à droite d'un tableau d'un certain nombre de positions.

```
Algorithme rotation
Variables :
        tab_n = [2, 15, 31, 17];
        tab_r[] : list(Tab);
        i : entier;

Début
        POUR i = 1  DANS tab_n FAIRE
                i = i % taille(tab_n);
                tab_r =  tab_n[-fin_indice(i)] + tab_n[-i];
        FINPOUR

        Afficher (“le tableau après la rotation est : “,tab_r);
Fin
```

Exercice 4 :
Écrire un programme qui compte le nombre de voyelles dans une chaîne de caractères présente dans un tableau.

```
Algorithme voyelle
Variables :
        nbre_voyelle, i : entier;
        tab = ['b','o','n','s','o','i','r'];

Début
        nbre_voyelle = 0;
        POUR i = 1 JUSQU'À 7 FAIRE
                si (tab[i] = "a" ou tab[i] = "e" ou tab[i] = "i" ou tab[i] = "o" ou tab[i] = "u") alors
                        nbre_voyelle = nbre_voyelle + 1;
                finsi
        FINPOUR
        ecrire ("le nombre de voyelles est : ",nbre_voyelle);
Fin
```

Exercice 5:
Écrire un programme qui trouve les éléments communs à deux tableaux d'entiers.

```
Algorithme éléments communs
Variables :
        tab1: Tab[];
        tab2: Tab[];
        n, i,y : entier;

Début
        Afficher(“la taille des deux tableaux”);
        lire (n);
        Afficher ("veuillez remplir le tab1");
        POUR i = 1 JUSQU'À n FAIRE
           lire tab1[i];
        FINPOUR

        Afficher ("veuillez remplir le tab2")
        POUR y = 1 JUSQU'À n FAIRE
           lire tab2[y];
        FINPOUR

        ecrire "Les éléments communs sont : "
        POUR i = 1 JUSQU'À n FAIRE
                POUR y = 1 JUSQU'À n FAIRE
                        SI tab1[i] = tab2[y] ALORS
                        Afficher (tab1[i]);
                        FINSI
                FINPOUR
        FINPOUR
Fin
```

Exercice 6 :
Écrire une fonction qui retourne tous les éléments qui n'ont pas de doublons dans un tableau.

```
Algorithme unique
Variables :
        mon_tableau : Tab[];
        resultats : Tab[]
        i : entier;

Début
        Fonction unique(tableau)
                elements_uniques : Tab[];
                Pour i dans tableau faire
                        si
        Fin Fonction
Fin
```

Exercice 7 :
Écrire un programme qui prend un tableau de chaînes de caractères et crée une nouvelle chaîne en concaténant tous les éléments.

```
Algo concaténation
Variables :
        tab[] : tableau ;
        n, i : entier ;
        nouvelle_chaine : chaine de caractère
Début
        afficher (« la taille de votre tableau de chaine de caractère ») ;
        lire (n) ;
        POUR i = 1 JUSQU'À n FAIRE
                afficher ("chaîne numéro : ",i)
                lire (tab[i])
        FINPOUR
        nouvelle_tab = « » ;
        POUR i = 1 JUSQU'À n FAIRE
                nouvelle_tab = nouvelle_tab + « »+ tab[i] ;
        FINPOUR
        afficher(nouvelle_tab) ;
Fin
```

Exercice 8 :
Écrire une fonction qui calcule la médiane d'un tableau d'entiers.

```
Algorithme médiane
Variables :
        n : entier ;

Début
        Fonction mediane(Tab)
        Tri(Tab)
        Si n%2 = 0, alors
                med ← (Tab[n/2 − 1] + Tab[n/2])/2
        Sinon
                med ← Tab[(n−1)/2]
        FinSi
        Retourner med
        Fin Fonction
Fin
```

Exercice 9 :
Écrire un programme qui calcule la fréquence de chaque caractère dans une chaîne de caractères.

Algorithme Occurrence
Variables :
tab_c[] : tableau de chaîne de caractères;

Exercice 10 :
Développer un programme pour déterminer si un patient est atteint de la COVID-19 implique généralement la vérification de certains symptômes ou facteurs de risque.