

DEVOIR DE CONTRÔLE Programmation en C

Tous documents interdits
Téléphones et calculatrices interdits
Soignez vos présentations et commentez vos programmes.

Exercice N°1 : 3 points

Écrire un programme C donnant en retour la somme des entiers pairs inférieurs à N .
 N est un nombre entier donné par l'utilisateur.

Exercice N° 2 : 5 points

Écrire un programme C qui lit N caractères au clavier et détermine :

- ✓ Le nombre de lettres minuscules.
- ✓ Le nombre de lettres majuscules.
- ✓ Le nombre de chiffres.
- ✓ Le nombre des autres caractères.
- ✓ Affiche les statistiques ainsi obtenues avant la sortie du programme.

Le nombre N de caractères saisis au clavier sera fixé par l'utilisateur.

On suppose que les codes ASCII des lettres se suivent de 'a' à 'z' et de 'A' à 'Z'.

Exercice N°3 : 6 points

Écrire un programme C qui calcule et affiche les $n^{\text{èmes}}$ termes des suites entières U_n et V_n définies ci-dessous.

$$\begin{cases} U_0 = 1 \\ U_n = V_{n-1} + 1 \end{cases} \quad \text{Et} \quad \begin{cases} V_0 = 0 \\ V_n = 2 * U_{n-1} \end{cases}$$

Pour une valeur de n inférieure et égale à 100 donnée par de l'utilisateur.

Exercice N°4 : 6 points

Écrire un programme C se comportant comme une calculatrice, c'est-à-dire exécutant une boucle sur :

- ✓ La lecture d'une ligne supposée contenir un entier, un opérateur et un entier (ex 1+3).
Les caractères représentant les opérateurs à considérer sont : +, -, *, % et /.
- ✓ Le calcul de la valeur de l'expression.
- ✓ L'impression du résultat à l'écran.

La sortie de la boucle d'exécution de ce programme doit être conditionnée par la volonté de l'utilisateur à poursuivre l'exécution du programme ou pas ceci en répondant à la question :

" Autre expression à évaluer O/N :? "

Programmation en C

Tous documents interdits
Téléphones et calculatrices interdits
Soignez vos présentations et commentez vos programmes.

Exercice N°1 (DC 29/10/2016)

Écrire un programme C qui demande à l'utilisateur d'introduire N rangs de plusieurs lettres de l'alphabet et à chaque fois, affiche la lettre relative au rang courant en minuscule et son code ASCII puis la même lettre en majuscule et son code ASCII.

Sachant :

- ✓ Qu'on considère que les rangs des lettres de l'alphabet s'étendent de 0 à 25. 0 est le rang de la lettre 'a', 25 est le rang de la lettre 'z'.
- ✓ Que les codes ASCII des lettres de l'alphabet se suivent de 'a' ... 'z' et de 'A' à 'Z' et que le code ASCII de la lettre 'A' est 65 et celui de la lettre 'a' est 97.

Exercice N°2 (DC 29/10/2016)

L'algorithme de Newton basé sur la convergence de la suite définie par :

$$U_0 = 1, \quad U_{n+1} = \frac{U_n + \frac{a}{U_n}}{2}$$

Converge vers \sqrt{a} .

Écrire un programme C qui exploite cet algorithme pour calculer la racine carrée d'un nombre réel positif ou nul donné par l'utilisateur.

Indications :

- ✓ La fonction prédéfinie $fabs(a)$ retourne la valeur absolue d'un réel. Elle est définie dans la bibliothèque `< math.h >`
- ✓ On suppose que la suite converge lorsque l'écart entre ses deux derniers termes sont suffisamment proches : $(fabs(U_{n+1} - U_n))$ de l'ordre de : $fabs(a) * 1.e^{-10}$.

Exercice N°3 (DC 10/03/2016)

Écrire un programme C qui lit plusieurs valeurs entières successives au clavier et affiche "paire" si elle est paire et "impaire" sinon.

EXAMEN

Programmation en C

Tous documents interdits
Téléphones et calculatrices interdits
Soignez vos présentations et commentez vos programmes.

Exercice N°1 : 7 points

Travail demandé :

1. Écrivez une fonction C qui inverse une chaîne de caractères (*ch1[]*) et met le résultat dans une autre (*ch2[]*). Les chaînes sont données en argument.

Prototype :

void Inverse_chaine(char ch1[], char ch2[]);

2. Écrivez une fonction C qui recherche dans une chaîne *ch* chaque caractère *car1* pour le remplacer par un caractère *car2*.

Prototype :

void cherche_remplace(char ch[], char car1, char car2);

3. Écrivez un programme C utilisant les fonctions précédentes qui réalise les actions suivantes :
 - a. Lecture au clavier d'une chaîne de caractères contenant un nombre variables de mots.
 - b. Inversion de cette chaîne de caractères dans une nouvelle chaîne.
 - c. Affichage de la chaîne initiale puis la chaîne inversée.
 - d. Recherche un caractère donné au clavier dans la chaîne initiale puis le remplace par un deuxième caractère donné également au clavier puis affiche la chaîne modifiée.

Exercice N° 2 : 13 points

Travail demandé :

1. Deux nombres sont opposés si leur somme est égale à 0. Deux nombres sont inverses si leur produit est égal à 1.

Écrivez une fonction C dont le prototype :

Int sontInvOuOpp(int a, int b);

Où *a* et *b* sont deux nombres entiers, cette fonction retourne 0 si *a* et *b* sont opposés, 1 si *a* et *b* sont inverses, autre valeur de votre choix sinon.

2. Écrivez une fonction C dont le prototype :

Int existeInvOuOppConsecutifs(int tab[], int nb);

Où **tab** est un tableau contenant **nb** nombres entiers. Cette fonction retourne **1** si **tab** contient deux nombres consécutifs opposés ou inverses, **0** sinon.

3. Écrivez une fonction C dont le prototype :

Void Echange(int * ptr1, int * ptr2);

Qui échange le contenu de deux variables entières ;

4. Écrivez une fonction C dont le prototype :

int indicemin(int tab[], int nb, int debut, int fin);

Qui retourne l'indice du plus petit élément parmi les éléments d'indices **debut** à **fin**.

Le tableau **tab** est un tableau contenant **nb** entiers. Lorsqu'il y a plusieurs minima, on prendra le premier trouvé

5. Le principe du tri par sélection est d'échanger le minimum du tableau avec le premier élément du tableau. Le plus petit élément est ainsi en tête. On recommence la même chose à partir du deuxième élément, puis du troisième, etc...

Écrire la fonction C dont le prototype :

void tri(int tab[], int nb);

Qui réalise le tri des éléments du tableau **tab** de **nb** entiers de la manière suivante :

Initialement, **debut** contient l'indice du second élément du tableau, **fin** est l'indice du dernier élément du tableau. **debut** et **fin** indiquent la partie du tableau non encore triée. Initialement, seul le premier élément est trié.

On itère **nb - 2** fois les instructions suivantes :

On recherche le plus petit élément du tableau entre les indices **debut** et **fin** (c'est le tableau non encore trié) ;

On échange cet élément avec le premier élément du tableau non encore trié ;

On incrémente **debut**, car un élément de plus est trié.

6. Écrivez la fonction C dont le prototype :

void RemplirTab(int tab[], int * ptrnb);

Qui remplit un tableau **tab** d'entiers à parti du clavier et sa taille effective.

7. Écrivez une fonction C dont le prototype :

void affiche(int tab[], int nb);

Qui affiche les **nb** éléments du tableau **tab** d'entiers à l'écran.

8. Écrivez un programme C utilisant les fonctions précédentes qui réalise les actions suivantes :

- Remplissage d'un tableau d'entiers et de sa taille.
- Teste si ce tableau d'entiers contient deux nombres consécutifs opposés ou inverses.
- Tri ce tableau d'entiers par la méthode de tri par sélection du minimum.
- Enfin affiche ce tableau.