جامعة قرطاج
Université de Carthage

République Tunisienne
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université de Carthage
École Nationale des Sciences et Technologies Avancées à Borj Cédria

ENSTAB

Date : 05/01/2022 **Durée :** 1h30	**Session :** ☒ Principale ☐ Rattrapage	**Documents :** ☒ non autorisés ☐autorisés **Calculatrice :** ☒ non autorisée ☐autorisée	**Nbr. de pages : 3**
Matière : Algorithmique Avancée et Programmation **Enseignant(es) :** H. Gharsellaoui, I. Msadaa, T. Bchini, S. Elghoul			

Exercice 1 : (5 pts)

Ecrire un programme qui multiplie 2 entiers positifs a et b selon le principe récursif suivant.

```
a * b = a * (b-1) + a     si b est impair
a * b = (2 * a) * (b/2)   si b est pair et différent de 0

EXEMPLE : 36 * 7 = 36 * 6 + 36
                 = 72 * 3 + 36
                 = 72 * 2 + 108
                 = 144 * 1 + 108
                 = 144 * 0 + 252
                 = 252
```

Ecrire un programme qui lit 2 entiers positifs a et b a partir du clavier et affiche leur produit selon l'algorithme récursif défini ci-dessus. Fournir les résultats tels qu'ils figurent dans l'exemple.

Exercice 2 : (6 pts)

On se propose dans cet exercice de calculer et d'afficher le montant total d'une facture.

Tous les prix sont exprimés en dinars et sont de type float.

La facture correspond à la somme des prix des produits en incluant la taxe, et les frais de livraison.

1. Ecrire la fonction ***taxeProduit()*** qui reçoit en argument le prix en hors taxe et lui rajoute le montant de la taxe. La taxe est de 10% du prix hors taxe pour les produits dont le prix est inférieur à 30 dinars et de 15% sinon. Penser à utiliser le passage par adresse pour cette fonction.
2. Ecrire la fonction ***fraisLivraison()*** qui reçoit en argument le montant d'une facture et retourne le montant des frais de port qui sont de 7 dinars si le montant est inférieur à 100 dinars sinon, la livraison est gratuite.

3. Déclarer une variable globale Tab, un tableau de 100 éléments correspondants aux prix hors taxe (HT) des produits et écrire une fonction *saisle()* qui permet de saisir les éléments de Tab. Faire un contrôle de saisie pour s'assurer que les prix sont positifs.
4. Au niveau du programme principal, appeler les fonctions définies précédemment pour :
    a. remplir Tab,
    b. calculer le montant total de la facture correspondant à la somme des prix TTC des produits
    c. calculer et afficher le montant de la facture (à 3 chiffres après la virgules) incluant les frais de port.

Exercice 3 : (9 pts)

Notez Bien :

Pour cet exercice, vous aurez besoin des fonctions ***isupper()*** et ***islower()*** de la bibliothèque ***<ctype.h>***

intislower(char caractere);

intisupper(char caractere);

islower() retourne une valeur non nulle si le caractère passé en argument est minuscule, et 0 sinon.

isupper() retourne une valeur non nulle si le caractère passé en argument est majuscule, et 0 sinon.

Exemple :

- islower('A') retourne 0 et islower('s') retourne un entier non nul.
- isupper('e') retourne 0 et isupper('A') retourne un entier non nul.

L'administrateur d'un site web désire écrire un programme qui lui permet d'évaluer la force des mots de passe des utilisateurs du site.

Pour cela, un score est attribué à chaque mot de passe.

- Le mot de passe est considéré « très faible » si le score<20
- Si 20<=score<40, le mot de passe est « faible »
- Si 40<=score<80 ; le mot de passe est « fort »
- Sinon le mot de passe est « très fort ».

Le score est calculé à travers la somme des bonus et des pénalités.

- Si le mot de passe est entièrement en minuscules, la pénalité est de -10.
- Si le mot de passe est entièrement en majuscules, la pénalité est de -5.

- Si la plus longue séquence de minuscules consécutives est < 3 caractères, le bonus est de 20.
- Si la plus longue séquence de majuscules consécutives est < 3 caractères, le bonus est de 30.
- On ajoute au bonus la longueur du mot de passe multipliée par 3.

1. Ecrire la fonction***sequenceMaj()***, qui reçoit en argument une chaîne de caractères et sa longueur et retourne la longueur de la séquence de majuscules la plus longue.
   intsequenceMaj(char*, int) ;
2. Ecrire la fonction ***sequenceMin()***, qui reçoit en argument une chaîne de caractères et sa longueur et retourne la longueur de la séquence de minuscules la plus longue.
   intsequenceMin(char*, int) ;
3. Ecrire la fonction ***bonus()*** qui reçoit en argument la chaîne de caractères et sa longueur et retourne la valeur du bonus en faisant appel aux autres fonctions.
4. Ecrire une fonction ***checkMaj()*** qui reçoit en argument la chaîne de caractères et sa longueur. Si la chaîne est entièrement composée de majuscules, la fonction retourne 1 sinon elle retourne 0.
5. Ecrire une fonction ***checkMin()*** qui reçoit en argument la chaîne de caractères et sa longueur. Si la chaîne est entièrement composée de minuscules, la fonction retourne 1 sinon elle retourne 0.
6. Ecrire la fonction ***penalites()*** qui reçoit en argument la chaîne de caractères et sa longueur et retourne la valeur de la pénalité en faisant appel aux fonctions qui entrent dans le calcul des pénalités.
7. Ecrire la fonction ***forcePwd()*** qui reçoit en argument le score et affiche le résultat : « fort », « faible », « très fort », « très faible »
8. Au niveau du programme principal, saisir un mot de passe, calculer son score en faisant appel aux fonctions ***bonus()*** et ***penalites()*** puis affiche l'évaluation de sa force en faisant appel à ***forcePwd()***.
9. Proposer un prototype et écrire une fonction ***empowerPwd()*** qui permet de renforcer un mot de passe en remplaçant chaque caractère 'i' qu'il contient par '!' et chaque 'a' par des '@'.
   La fonction affiche le résultat obtenu (le nouveau mot de passe éventuellement transformé).

Bon courage et bonne chance