Partie I : Théorique (11 pts)

1. Un compilateur permet de :
   * Vérifier la syntaxe du programme.
   * Traduire le programme en langage machine.
2. Quelle est la différence entre le passage de paramètres d’une fonction par adresse et le passage de paramètres par valeur ? donner un exemple :
   * Dans le passage par valeur : la fonction utilise une copie des paramètres effectifs. Après appel de la fonction les paramètres effectifs gardent leurs valeurs initiales.
   * Dans le passage par adresse : la fonction utilise les adresses des paramètres effectifs. Alors ces paramètres peuvent changer après appel de la fonction
   * Exemple : Fonction Permuter
3. Citer quatre fonctions de la bibliothèque **string.h** :
   * **strlen**: renvoie la longueur d’une chaine
   * **strcpy**: copie une chaine dans une autre
   * **strcmp**: compare deux chaines
   * **strcat**: concatène deux chaines
4. Que sera le résultat d’exécution des programmes suivants :

* **Programme 1 :**
* **int** i = 10;
* for(i = 0;i < 10; i++)

printf("%d\n", i);

**Résultat d’exécution :**

0  
1  
2  
3  
4  
5  
6  
7  
8  
9

* **Programme 2 :**

**int** i = 0;   
 while(i < 10);  
 i++;  
 printf("%d\n", i);

**Résultat d’exécution :**

Boucle infinie!!

**Notes**

* + Notez bien la présence d'un point-virgule à la fin des parenthèses de la boucle while, ce qui produira une boucle infinie puisque i sera toujours inférieur à 10.
  + Sans le point-virgule, ce programme affichera le nombre 10.
* **Programme 3 :**

**int** i = 0, j = 10;   
 do{  
 printf("%d\n", i);   
 i++;  
 j = i + j;  
 }while((i < 10) && (j <20));

**Résultat d’exécution :**

0  
1  
2  
3

* **Programme 4 :**

**int** i = 1;   
 switch(i){  
 case 0 : puts("0");  
 case 1 : puts("1");  
 case 2 : puts("2");   
 default : puts("Good By");  
 }

**Résultat d’exécution :**

1  
2  
Good By

**Note**

Notez bien l'absence de l'instruction **break** donc ce programme ne s’arrêtera pas dans le case 1.

Partie II : Pratique (29 pts)

Exercice 1 : (5 pts)

1. //Auteur: IDMANSOUR
2. //Copyright: Exelib.net
4. #include<stdio.h>
5. **int** suite(**int** n){
6. if(n == 1 || n == 2)
7. return 1;
9. return 2 \* (suite(n-1) + suite(n-2));
10. }
12. main(){
13. **int** nb, i, t, s = 0;
15. printf("Entrer le nombre de termes: ");
16. scanf("%d", &nb);
17. for(i = 1; i <= nb ; i++){
18. t = suite(i);
19. printf("U(%d) = %d\n", i, t);
20. s = s + t;
21. }
23. printf("\nLa somme des termes : %d", s);
24. }

**Exemple d’exécution :**

Entrer le nombre de termes: 5  
U(1) = 1  
U(2) = 1  
U(3) = 4  
U(4) = 10  
U(5) = 28

La somme des termes : 44

Exercice 2 : (24 pts)

1. //Auteur: IDMANSOUR
2. //Copyright: Exelib.net
4. #include<stdio.h>
5. #include<string.h>
7. //Question 1 : Définition de la structure Article
8. typedef struct {
9. **int** code;
10. char nom[20];
11. **float** prix;
12. } **Article**;
14. //Question 2 : Lecture des articles
15. void lireArticle (**Article** articles [], **int** n) {
16. **int** i;
17. for(i=0; i<n; i++){
18. printf("Article %d : \n", i+1);
19. printf("Entrer le code : ");
20. scanf("%d", &articles[i].code);
21. printf("Entrer le nom : ");
22. scanf("%s", articles[i].nom);
23. printf("Entrer le prix : ");
24. scanf("%f", &articles[i].prix);
25. }
26. }
28. //Question 3 : Affichage de la liste des articles
29. void afficherArticle (**Article** articles [], **int** n) {
30. **int** i;
31. for(i=0; i<n; i++){
32. printf("Article %d : \n", i+1);
33. printf("Code : %d | ", articles[i].code);
34. printf("Nom : %s | ", articles[i].nom);
35. printf("Prix : %.2f\n\n", articles[i].prix);
36. }
37. }
39. //Question 4 : Suppression d'un article
40. void supprimerArticle (**Article** articles [], **int** \* n, **int** code) {
41. **int** i, j;
42. //On suppose si le code est unique
43. for(i=0; i<\*n; i++){
44. if(articles[i].code == code){
45. for(j=i; j<(\*n)-1; j++){
46. articles[j] = articles[j+1];
47. }
49. (\*n)--;
50. //i--; si le code n'est pas unique
51. }
52. }
53. }
55. //Question 5 : Ajout d'un article à la fin de la liste
56. void ajouterArticle (**Article** articles [], **int** \*n){
57. //Ajout a la fin?
58. printf("Entrer le code : ");
59. scanf("%d", &articles[\*n].code);
60. printf("Entrer le nom : ");
61. scanf("%s", articles[\*n].nom);
62. printf("Entrer le prix : ");
63. scanf("%f", &articles[\*n].prix);
64. (\*n)++;
65. }
67. //Question 6 : Tri de la liste des articles par ordre (Croissant) alphabétique des noms
68. void triByNom (**Article** articles [], **int** n){
69. **int** i, j;
70. **Article** a;
71. for(i=0; i<n; i++){
72. for(j=i;j<n;j++){
73. if(strcmp(articles[i].nom , articles[j].nom) == 1)
74. {
75. a = articles[i];
76. articles[i] = articles[j];
77. articles[j] = a;
78. }
79. }
80. }
81. puts("Liste des articles tries par nom:");
82. afficherArticle(articles, n);
83. }
85. //Question 7 : Tri de la liste des articles du plus cher au moins cher (par ordre décroissant de prix)
86. void triByPrix (**Article** articles [], **int** n){
87. **int** i, j;
88. **Article** a;
89. for(i=0; i<n; i++){
90. for(j=i;j<n;j++){
91. if(articles[i].prix < articles[j].prix)
92. {
93. a = articles[i];
94. articles[i] = articles[j];
95. articles[j] = a;
96. }
97. }
98. }
99. puts("Liste des articles tries par prix:");
100. afficherArticle(articles, n);
101. }
103. //Question 8 : Sauvegrade de la liste des articles dans un fichier texte
104. void saveArtciles (**Article** articles [], **int** n){
105. **FILE** \*f;
106. **int** i;
107. f = fopen("articles.txt", "w");
108. if(!f){
109. printf("Erreur de creation du fichier!");
110. return;
111. }
112. for(i=0; i<n; i++){
113. fprintf(f, "%d;%s;%.2f\n", articles[i].code, articles[i].nom, articles[i].prix);
114. }
115. fclose(f);
116. }
118. //Question 9 : Programme de test
119. main(){
121. //Article t[30] = {{1, "dsw", 300}, {2, "xue", 30}, {3, "adf", 380}}, \* a;
122. **Article** t[30], \* a;
123. **int** n, choix, code;
124. do{
125. //Menu
126. puts("-------------Menu-------------");
127. puts("1: Lire les articles.");
128. puts("2: Afficher les articles classés par ordre alphabétique des noms.");
129. puts("3: Afficher les articles classés du plus cher au moins cher.");
130. puts("4: Ajouter un nouvel article.");
131. puts("5: Supprimer un article.");
132. puts("6: Enregistrer les articles.");
133. puts("7: Quitter le programme.");
134. puts("Tapez votre choix :");
135. scanf("%d", &choix);
137. switch(choix){
138. case 1:
139. printf("Donner le nombre d'articles: ");
140. scanf("%d", &n);
141. lireArticle(t, n);
142. break;
143. case 2:
144. triByNom(t,n);
145. break;
146. case 3:
147. triByPrix(t,n);
148. break;
149. case 4:
150. ajouterArticle(t, &n);
151. break;
152. case 5:
153. printf("Entrer le code de l'article a supprime: ");
154. scanf("%d", &code);
155. supprimerArticle(t, &n, code);
156. printf("Liste des articles apres suppression de l'article %d : \n", code);
157. afficherArticle(t, n);
158. break;
159. case 6:
160. saveArtciles(t, n);
161. printf("Liste sauvegardee dans le fichier articles.txt\n");
162. break;
163. case 7:
164. printf("Fin du programme\n");
165. break;
166. default:
167. printf("Choix invalid!\n");
169. }
170. }while(choix != 7);
171. }  
     //Source : www.exelib.net