

---

## Travaux pratiques 9 : Qu'y a-t-il au menu ?

---

**Correction.** Note aux chargés de TD.

- L'objectif du TP est de les familiariser avec les fonctions au travers d'un des points présenté en cours : la structuration du code. Ils créent un menu permettant l'exécution de différents sous-problèmes qu'il sont déjà résolus en TP, chaque sous-problème étant implanté dans une fonction.
- On incite ici les étudiants à structurer, indenter et commenter correctement leur code, sans quoi ils ne viendront pas à bout de ce type d'exercice.

Dans les TP précédents vous avez réalisé plusieurs programmes en C effectuant chacun une tâche. Le but de ce TP est de réunir plusieurs de ces programmes en un seul, dans lequel l'utilisateur choisira la tâche à effectuer dans un menu. À la fin de l'exécution d'une tâche, le menu est à nouveau affiché pour laisser le choix à l'utilisateur d'exécuter d'autres tâche ou de quitter le programme. Un exemple d'exécution est donné plus bas.

### 1 Un menu

Dans un répertoire TP9, écrire et tester le programme `menu1.c` de manière à ce que :

1. le programme affiche un menu proposant 3 choix représentés par des entiers : (1) tester si un entier est premier, (2) calculette ou (0) quitter. L'utilisateur fera son choix en entrant un entier.
2. Si cet entier est 0, mettre fin au programme.
3. Sinon : si cet entier est 1 afficher « premier : non disponible », si c'est 2, « calculette : non disponible » sinon « choix non disponible », puis boucler à l'étape 1.

### 2 Des programmes

- Dans votre répertoire TP9, créer le programme `premier.c` qui teste si un nombre entré par l'utilisateur est premier (voir TD). Vérifier que votre programme fonctionne.
- Dans votre répertoire TP6 (ou celui de votre binôme) doit se trouver le programme réalisant une mini-calculatrice. Vérifier que celui-ci est correctement indenté et commenté, et qu'il fonctionne.

### 3 Intégration des programmes dans le menu en utilisant les fonctions

Remarque : pour copier/coller sous un système unix vous pouvez : (copier) sélectionner le texte à copier à l'aide de la souris; (coller) effectuer un clic du milieu (bouton-molette) à l'endroit où vous souhaitez coller.

- Enregistrer le fichier `menu1.c` sous le nom `menu2.c`.
- Dans `premier.c`, extraire le code relatif au problème et intégrer-le dans une fonction qui sera appelée à la bonne place dans votre `menu2.c` : déclarer une fonction `int est_premier(int n)` et la définir. Cette fonction renverra la constante symbolique `TRUE` si  $n$  est premier et `FALSE` (valeur 0) sinon. Faire en sorte que le traitement du choix 1 de l'utilisateur utilise cette fonction pour déterminer si un nombre est premier.
- Faire la même chose avec la calculatrice.
- Tester `menu2.c` et s'il vous reste du temps ajouter des choix dans le menu, inspirés par les différents problèmes que vous avez déjà résolus dans les TP précédents ("factorielle", "deviner le nombre secret", "sortir le chien", "majeure, mineure").

<pre>***** MENU ***** * * 1) Tester si un nombre est premier * * 2) Calculatrice * * 0) QUITTER * * ***** votre choix : 1 Donner un nombre entier positif : 34 Le nombre 34 n'est pas premier, 2 divise 34 ***** MENU ***** * * 1) Tester si un nombre est premier * * 2) Calculatrice * * 0) QUITTER * * ***** votre choix : 2</pre>	<pre>Entrer expression : nombre operateur nombre 4.001 * 0.7 4.001 * 0.7 = 2.8007  ***** MENU ***** * * 1) Tester si un nombre est premier * * 2) Calculatrice * * 0) QUITTER * * ***** votre choix : 0  Sayonara</pre>
---	---

**Correction.** Correction avec d'autres fonctions ajoutées au menu. La correction est légèrement différente de ce qui est demandé.

```
1 /* -*- coding: utf-8 -*- pour que emacs travaille en utf8 */
2 /* menu3.c (L1 EI), debut de l'utilisation de fonctions dans le menu */
3 #include <stdlib.h> /* EXIT_SUCCESS, rand, srand */
4 #include <stdio.h> /* printf, scanf */
5 #include <time.h> /* time */
6
7 #define TRUE 1
8 #define FALSE 0
9 #define NB_MAX 100
10 /* declaration de fonctions utilisateurs */
11
12 /* test de primalité */
13 int est_premier(int n);
```

```

14
15 /* Calcul de la factorielle (vaut 1 si l'argument est négatif) */
16 int factorielle(int n);
17
18 /* Tirer un nombre au hasard entre 0 et n */
19 int nombre_aleatoire(int n);
20
21 /* motif cercle de rayon 'rayon' et de centre (0,0) */
22 char cercle(int x, int y, int rayon);
23
24
25 int main()
26 {
27     int continuer = TRUE; /* TRUE si on doit proposer le menu */
28     int choix_menu = 0; /* Choix de l'utilisateur */
29
30     srand(time(NULL)); /* à ne faire qu'une fois */
31
32     /* Boucle principale d'interaction avec l'utilisateur */
33     while(continuer)
34     {
35         /* Affichage du menu */
36         printf("\n\n");
37         printf("***** MENU *****\n");
38         printf("*                               *\n");
39         printf("* 1) Tester si un nombre est premier *\n");
40         printf("* 2) Calculette                               *\n");
41         printf("* 3) Deviner un nombre                               *\n");
42         printf("* 4) Cercle d'etoiles                               *\n");
43         printf("* ...                               *\n");
44         printf("*                               *\n");
45         printf("*                               *\n");
46         printf("* 0) QUITTER                               *\n");
47         printf("*                               *\n");
48         printf("***** votre choix : ");
49
50         /* Choix utilisateur */
51         scanf("%d", &choix_menu);
52
53         /* Execution du choix de l'utilisateur (cas mutuellement exclusifs) */
54
55         if (1 == choix_menu) /* ----- Test de primalité -----
56         {
57             int p; /* nombre a tester */
58
59             /* Saisie utilisateur */
60             printf("Donner un nombre entier positif : ");

```

```

61         scanf("%d", &p);
62
63         if (est_premier(p)) /* p est premier ... ou négatif */
64         {
65             printf("Le nombre %d est premier\n", p);
66         }
67         else /* x n'est pas premier et i - 1 divise x */
68         {
69             printf("Le nombre %d n'est pas premier\n", p);
70         }
71     }
72
73     if (2 == choix_menu) /* ----- Calculette ----- */
74     {
75         double nombre_g; /* membre gauche de l'expression */
76         double nombre_d; /* membre droit de l'expression */
77         char op; /* operateur */
78         double expr; /* resultat de l'expression */
79
80         /* saisie expression */
81         printf("Entrez une expression de la forme : nombre operateur nombre\n");
82         scanf("%lg",&nombre_g);
83         scanf(" %c",&op);
84         if (op == '!')
85         {
86             int n = nombre_g; /* ignorer la décimale */
87             expr = factorielle(n);
88             nombre_g = n; /* ignorer la décimale dans l'affichage */
89         }
90         else
91         {
92             scanf("%lg",&nombre_d);
93         }
94         /* calcul valeur expression */
95         /* cas mutuellement exclusif */
96         if(op == '+') /* addition */
97         {
98             expr = nombre_g + nombre_d;
99         }
100
101         if(op == '-') /* soustraction */
102         {
103             expr = nombre_g - nombre_d;
104         }
105
106         if(op == '*') /* multiplication */
107         {

```

```

108         expr = nombre_g * nombre_d;
109     }
110
111     if(op == '/') /* division */
112     {
113         expr = nombre_g / nombre_d;
114     }
115
116     /* affichage resultat */
117     if (op == '!')
118     {
119         printf("%g! = %g\n",nombre_g, expr);
120     }
121     else
122     {
123         printf("%g %c %g = %g\n",nombre_g,op,nombre_d,expr);
124     }
125 }
126
127 if (3 == choix_menu) /* ----- Deviner un nombre -----
128 {
129     int choix; /* choix de l'utilisateur pour le nombre secret */
130     int trouve = FALSE; /* TRUE si trouvé */
131     int nombre_secret;
132
133     /* Tirage aléatoire du nombre secret */
134     nombre_secret = nombre_aleatoire(NB_MAX);
135
136     /* manche joueur */
137     while(!trouve) /* pas trouvé nombre secret */
138     {
139         /* demande nombre à l'utilisateur */
140         printf("Votre choix (nombre entre 0 et %d) ?\n",NB_MAX);
141         scanf("%d",&choix);
142
143         if(choix == nombre_secret) /* trouvé */
144         {
145             trouve = TRUE;
146         }
147         else /* pas trouvé */
148         {
149             /* donne indice */
150             if(choix > nombre_secret)
151             {
152                 printf("Trop grand.\n");
153             }
154             else

```

```

155         {
156             printf("Trop petit.\n");
157         }
158     }
159 }
160 /* trouvé nombre secret */
161
162     printf("Vous avez trouvé le nombre secret.\n");
163 }
164
165 if (4 == choix_menu) /* ----- Cercle -----
166 {
167     int ligne; /* numero de ligne */
168     int colonne; /* numero de colonne */
169     int rayon; /* rayon du cercle */
170
171     printf("Donner le rayon : ");
172     scanf("%d", &rayon);
173     printf("\n");
174
175     /* Affichage par balayage */
176     for (ligne = -rayon; ligne <= rayon; ligne = ligne + 1)
177     {
178         for (colonne = -rayon; colonne <= rayon; colonne = colonne + 1)
179         {
180             printf("%c", cercle(colonne, ligne, rayon));
181         }
182         printf("\n");
183     }
184 }
185 }
186
187 if (4 < choix_menu) /* Non disponible */
188 {
189     printf("\n** Choix non disponible **\n");
190 }
191
192
193 if (0 < choix_menu) /* Attendre que l'utilisateur soit pret a revenir au m
194 {
195     char c;
196     printf("\n[Saisir un caractere pour revenir au menu] ");
197     scanf(" %c", &c); /* un caractère blanc ne sera pas pris en compte */
198 }
199
200 if (0 == choix_menu) /* quitter */
201 {

```

```

202         printf("Sayonara\n");
203         continuer = FALSE;
204     }
205 }
206
207     /* valeur fonction */
208     return EXIT_SUCCESS;
209 }
210
211 /* implantation de fonctions utilisateurs */
212 int est_premier(int n)
213 {
214     int i;
215
216     for (i = 2; i < n; i = i + 1)
217     {
218         if (n % i == 0)
219         {
220             return FALSE;
221         }
222     }
223     return TRUE;
224 }
225
226
227 int factorielle(int n)
228 {
229     int i; /* Var. de boucle */
230     int res = 1; /* resultat */
231
232     for (i = 1; i <= n; i = i + 1) /* Pour i = 1, 2, ..., n */
233     {
234         res = res * i; /* mettre i dans le produit */
235     }
236
237     /* Valeur fonction */
238     return res;
239 }
240
241 char cercle(int x, int y, int rayon)
242 {
243     if (x*x + y*y <= rayon * rayon)
244     {
245         return '*';
246     }
247     return ' ';
248 }

```

```
249
250 int nombre_aleatoire(int n)
251 {
252     /* tirage du nombre secret */
253     return rand() % n; /* entre 0 et n inclus */
254 }
```