

Travaux dirigés 5 : fonctions et procédures (2)

Durant les séances précédentes vous avez réalisé plusieurs programmes en C effectuant chacun une tâche. Le but ici est d'utiliser des fonctions pour commencer à 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 tâches ou de quitter le programme. Un exemple d'exécution est donné plus bas.

Comme vous allez écrire un gros programme dans un seul fichier source, il est essentiel que vous structuriez bien votre fichier source, et que celui-ci soit correctement indenté. Si vous n'aimez pas gérer vous-même l'indentation apprenez à utiliser l'éditeur de texte **emacs** et son indentation automatique.

Vous travaillerez sur trois parties du programme :

- les déclarations du début du programme (fonctionnalités, constantes, fonctions),
- les définitions de fonctions,
- la fonction principale (**main**).

Vous ferez en sorte de pouvoir **tester votre programme le plus tôt possible et le plus souvent possible**, quitte à afficher à l'utilisateur que certaines choses ne sont pas disponibles / pas terminées.

<pre>***** MENU ***** * * 1) Tester si un nombre est premier * * 2) Deviner un nombre * * 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) Deviner un nombre * * 0) QUITTER * * ***** votre choix : 2 Votre choix ? 8</pre>	<pre>Plus petit. Votre choix ? 4 Plus petit. Votre choix ? 2 Vous avez trouvé le nombre secret. ***** MENU ***** * * 1) Tester si un nombre est premier * * 2) Deviner un nombre * * 0) QUITTER * * ***** votre choix : 0 Sayonara</pre>
---	--

1. Commencer par faire en sorte que :

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

Il est recommandé de découper ce programme en fonctions et procédures.

- Déclarer et définir une procédure **afficher_menu()** qui affichera le menu. Tester là.

Correction. _____ *C'est tout bête.*

```
/* Déclarations de fonctions utilisateurs */
void afficher_menu();
...
```

```

/* Definitions de fonctions utilisateurs */
void afficher_menu()
{
    printf("***** MENU *****\n");
    printf("*\n");
    printf("1) Tester si un nombre est premier *\n");
    printf("2) Calcul de factorielle *\n");
    printf("3) Deviner un nombre *\n");
    printf("4) Motif d'etoiles *\n");
    printf("*\n");
    printf("0) QUITTER *\n");
    printf("*\n");
    printf("***** votre choix :");
}

```

- Déclarer et définir une fonction `choix_utilisateur()`, sans paramètres qui renverra une valeur entière saisie par l'utilisateur. Tester là.

Correction.

```

int choix_utilisateur()
{
    int choix;
    scanf("%d", &choix);
    return choix;
}

```

- Déclarer et définir une fonction `executer_menu()` qui :
 - affichera le menu à l'utilisateur et réalisera la saisie de son choix (avec les fonctions et procédures précédentes);
 - lorsque ce choix est 1, appellera une procédure `menu_premier()` (pendant la mise au point du programme, faites en sorte que cette procédure affiche "non disponible");
 - puis, lorsque ce choix est différent de 0, renverra `TRUE` et lorsque ce choix est égal à zéro renverra `FALSE`.

Correction. _____ *Les étudiants doivent se poser la question de ce dont pourrait avoir besoin `menu_premier` pour effectuer le traitement et ce qu'elle doit renvoyer (réponse : rien dans les deux cas).*

```

int executer_menu()
{
    int choix;

    /* Affichage du menu et choix de l'utilisateur */
    afficher_menu();
    choix = choix_utilisateur();

    if (1 == choix) /* ----- 1) Tester si un nombre est premier ----- */
    {
        menu_premier();
    }

    /* Valeur de retour */
    if (choix != 0)
    {
        return TRUE;
    }
    return FALSE;
}

```

- Modifier la fonction `main` de telle sorte qu'elle fasse appel à la fonction `executer_menu` tant que celle-ci renvoie `TRUE`.

Correction.

```

int main()
{

```

```

/* Declarations et initialisation des variables */
int encore = TRUE;

/* Boucle d'interaction avec l'utilisateur */
while (encore)
{
    encore = executer_menu();
}

/* Greetings */
printf("Bye bye\n");

/* Valeur fonction */
return EXIT_SUCCESS;
}

```

1 Ajouts de fonctionnalités

Vous pouvez maintenant commencer à programmer quelques unes des tâches que propose le menu. Si vous possédez des programmes réalisant ces tâches, inspirez vous en (copier/coller).

2. Redéfinir la procédure `menu_premier()` de manière à tester si un nombre choisi par l'utilisateur est premier. Elle fera appel à une fonction `est_premier()` (dont vous retrouverez la définition et la déclaration dans le TD 4) et à la fonction `choix_utilisateur()` pour le choix de l'entier.

Correction.

```

void menu_premier()
{
    int p;

    printf("Donner un nombre : ");
    p = choix_utilisateur();

    if (est_premier(p))
    {
        printf("Le nombre %d est premier\n", p);
    }
    else
    {
        printf("Le nombre %d n'est pas premier\n", p);
    }
}

/* Rappel : */
int est_premier(int n)
{
    int i = 2;
    int premier = TRUE;

    while (premier && i < n)
    {
        if (n % i == 0)
        {
            premier = FALSE;
        }
        i = i + 1;
    }
    return premier;
}

```

3. Modifier votre fonction `choix_utilisateur()` de telle sorte que :
 - elle prenne en argument deux paramètres entiers a et b ;

- si l'utilisateur saisit un nombre $n \in [a, b]$ la fonction retourne n sans générer d'affichage;
- si l'utilisateur saisit un nombre $n \notin [a, b]$ l'intervalle de saisie est affiché à l'utilisateur et la saisie redemandée, jusqu'à cinq fois.
- si au bout de cinq fois l'utilisateur n'a toujours pas donné un nombre dans l'intervalle, la procédure renvoie a .
- Tester. Quel est le problème lorsque l'utilisateur saisie autre chose que des chiffres au clavier? Nous corrigerons ça un peu plus loin.

Correction.

```
int choix_utilisateur(int a, int b)
{
    int compteur = 5; /* compteur du nombre d'essais */
    int choix; /* choix de l'utilisateur */

    scanf("%d", &choix);
    while ((compteur > 0) && ((choix < a) || (choix > b)))
    {
        printf("Le nombre doit etre entre %d et %d (inclus) : ", a, b);
        scanf("%d", &choix);
        compteur = compteur - 1;
    }
    if ( (choix < a) || (choix > b) ) {
        choix = a;
    }
    return choix;
}
```

IL faut aussi modifier les appels à cette fonction. Pour la fonction `executer_menu()` on donne l'intervalle $[0, 3]$ (à supposer que ces quatre choix soient traités). Pour la procédure `menu_premier()`, on donne l'intervalle $[1, \text{INT_MAX}]$ (charger `limits.h` pour définir la constante symbolique `INT_MAX`, la plus grande valeur que peut prendre un entier). —

4. Ajouter le nécessaire à votre programme pour pouvoir jouer à deviner un nombre.

Correction. — *Il faut écrire une procédure `menu_deviner()`, qui fera appel à une fonction `tirage_aleatoire` puis en boucle à `choix_utilisateur` sur l'intervalle dans lequel le nombre est tiré (de 0 à une constante symbolique).*

Un code complet du programme à la fin du TD (mais personne ne va finir non ?) :

```
1  /* Fonctionnalités supplémentaires */
2  #include <stdlib.h> /* EXIT_SUCCESS, rand(), srand() */
3  #include <stdio.h> /* printf(), scanf() */
4  #include <limits.h> /* INT_MAX */
5  #include <time.h> /* time() */
6
7  /* Déclarations de types et constantes utilisateurs */
8  #define TRUE 1
9  #define FALSE 0
10 #define DEVINER_MAX 100
11 #define DEVINER_ESSAIS 8
12
13 /* Déclarations de fonctions utilisateurs */
14
15 /* Affiche le menu */
16 void afficher_menu();
17
18 /* Récupère un nombre entier saisi par l'utilisateur dans l'intervalle
19  * [a,b] et le retourne */
20 int choix_utilisateur(int a, int b);
21
22 /* Affichage du menu et traitement du choix de l'utilisateur */
23 int executer_menu();
24
25 /* Teste si un nombre saisi par l'utilisateur est premier et affiche le résultat */
26 void menu_premier();
```

```

27
28  /* Teste si son argument (positif) est premier */
29  int est_premier(int p);
30
31  /* Factorielle */
32  void menu_factorielle();
33  int factorielle(int n);
34
35  /* Jouer a une devinette */
36  void menu_deviner();
37
38  /* Tirer un nombre au hasard entre 0 et n */
39  int nombre_aleatoire(int n);
40
41  /* calculette */
42  void menu_calculatrice();
43  int calculatrice();
44
45  /* motif */
46  void afficher_motif(int cote);
47  int motif(int colonne, int ligne);
48
49  /* Fonction principale */
50  int main()
51  {
52      /* Declarations et initialisation des variables */
53      int encore = TRUE;
54
55      /* Initialisation du generateur aleatoire */
56      srand(time(NULL)); /* à ne faire qu'une fois */
57
58      /* Boucle d'interaction avec l'utilisateur */
59      while (encore)
60      {
61          encore = executer_menu();
62      }
63
64      /* Greetings */
65      printf("Bye bye\n");
66
67      /* Valeur fonction */
68      return EXIT_SUCCESS;
69  }
70
71  /* Definitions de fonctions utilisateurs */
72  void afficher_menu()
73  {
74      printf("***** MENU *****\n");
75      printf("*\n");
76      printf("*  1) Tester si un nombre est premier  *\n");
77      printf("*  2) Calcul de factorielle          *\n");
78      printf("*  3) Deviner un nombre                 *\n");
79      printf("*  4) Motif d'etoiles                   *\n");
80      printf("*  5) Calculatrice                       *\n");
81      printf("*\n");
82      printf("*  0) QUITTER                           *\n");
83      printf("*\n");
84      printf("***** votre choix : ");
85  }
86
87  int choix_utilisateur(int a, int b)
88  {
89      int compteur = 0; /* compteur du nombre d'essais */
90      int choix = 0; /* choix de l'utilisateur */
91      char saisie[256]; /* declaration d'un chaine qui nous servira de
92                      * "tampon" */
93      choix = a - 1;
94
95      while ((compteur < 5) && ((choix < a) || (choix > b)))

```

```

96     {
97         if (compteur > 0) {
98             printf("Le nombre doit etre entre %d et %d (inclus) : ", a, b);
99         }
100         fgets(saisie, 256, stdin); /* preleve la ligne tapee par
101                                     l'utilisateur et la place dans le
102                                     tampon */
103         sscanf(saisie, "%d", &choix); /* tente de reconnaitre un entier
104                                         dans le tampon */
105         compteur = compteur + 1;
106     }
107     if ((choix < a) || (choix > b)) {
108         choix = a;
109     }
110     return choix;
111 }
112
113 int executer_menu()
114 {
115     int choix;
116
117     /* Affichage du menu et choix de l'utilisateur */
118     afficher_menu();
119     choix = choix_utilisateur(0, 5);
120
121     if (1 == choix) /* ----- 1) Tester si un nombre est premier ----- */
122     {
123         menu_premier();
124     }
125
126     if (2 == choix) /* ----- 2) calcul de factorielle ----- */
127     {
128         menu_factorielle();
129     }
130
131     if (3 == choix) /* ----- 3) Deviner un nombre ----- */
132     {
133         menu_deviner();
134     }
135
136     if (4 == choix) /* ----- 4) Motif d'etoiles ----- */
137     {
138         afficher_motif(10);
139     }
140
141     if (5 == choix) /* ----- 5) Calculette ----- */
142     {
143         menu_calculatrice();
144     }
145
146     /* Valeur de retour */
147     if (choix != 0)
148     {
149         return TRUE;
150     }
151     return FALSE;
152 }
153
154 void menu_premier()
155 {
156     int p;
157
158     printf("Donner un nombre : ");
159     p = choix_utilisateur(1, INT_MAX);
160
161     if (est_premier(p))
162     {
163         printf("Le nombre %d est premier\n", p);
164     }

```

```

165     else
166     {
167         printf("Le nombre %d n'est pas premier\n", p);
168     }
169 }
170
171
172 int est_premier(int n)
173 {
174     int i = 2;
175     int premier = TRUE;
176
177     while (premier && i < n)
178     {
179         if (n % i == 0)
180         {
181             premier = FALSE;
182         }
183         i = i + 1;
184     }
185     return premier;
186 }
187
188 void menu_factorielle()
189 {
190     int n;
191     printf("Calcul de factorielle.\nEnter l'argument (entier positif) : ");
192     n = choix_utilisateur(0, INT_MAX);
193     printf("%d\n", factorielle(n));
194 }
195
196 int factorielle(int n)
197 {
198     int i;
199     int res = 1;
200     for (i = 2; i <= n; i = i + 1)
201     {
202         res = res * i;
203     }
204     return res;
205 }
206
207 void menu_deviner()
208 {
209     int choix; /* choix de l'utilisateur pour le nombre secret */
210     int trouve = FALSE; /* TRUE si trouvé */
211     int nombre_secret;
212     int essais = DEVINER_ESSAIS; /* essais restants */
213
214     /* Tirage aléatoire du nombre secret */
215     nombre_secret = nombre_aleatoire(DEVINER_MAX);
216
217     /* manche joueur */
218     while(!trouve && (essais > 0)) /* pas trouvé nombre secret */
219     {
220         /* demande nombre à l'utilisateur */
221         printf("Votre choix (nombre entre 0 et %d) : ", DEVINER_MAX);
222         choix = choix_utilisateur(0, DEVINER_MAX);
223
224         if(choix == nombre_secret) /* trouvé */
225         {
226             trouve = TRUE;
227         }
228         else /* pas trouvé */
229         {
230             /* donne indice */
231             if(choix > nombre_secret)
232             {
233                 printf("Trop grand.\n");

```

```

234         }
235         else
236         {
237             printf("Trop petit.\n");
238         }
239     }
240     essais = essais - 1;
241 }
242
243 if (essais > 0)
244 {
245     /* trouvé nombre secret */
246     printf("Gagné. Vous avez trouvé le nombre secret.\n");
247 }
248 else
249 {
250     /* Perdu */
251     printf("Perdu : limite du nombre d'essais atteinte.\n");
252 }
253 }
254
255
256 int nombre_aleatoire(int n)
257 {
258     /* tirage du nombre secret */
259     return rand() % (n + 1); /* entre 0 et n inclus */
260 }
261
262 void menu_calcullette()
263 {
264     int continuer = TRUE;
265     printf("Mode calcullette\n");
266     printf("Entrer des expressions nombre operateur nombre\n");
267     printf("ou 0+0 pour quitter\n");
268     while (continuer)
269     {
270         printf("Votre expression ? ");
271         continuer = calcullette();
272     }
273 }
274
275 int calcullette()
276 {
277     double x;
278     double y;
279     char op;
280
281     scanf("%lg %c %lg", &x, &op, &y);
282
283     if ('+' == op) /* addition */
284     {
285         if ((0 == x) && (0 == y))
286         {
287             printf("La tete a toto \n");
288             return FALSE;
289         }
290         printf("%lg\n", x + y);
291     }
292     else if ('-' == op) /* soustraction */
293     {
294         printf("%lg\n", y - x);
295     }
296     else if (('*' == op) || ('x' == op)) /* multiplication */
297     {
298         printf("%lg\n", x * y);
299     }
300     else if ( '/' == op) /* division */
301     {
302         printf("%lg\n", x / y);

```



```

303     }
304     else
305     {
306         printf("operation non reconnue\n");
307         return FALSE;
308     }
309     return TRUE;
310 }
311
312 void afficher_motif(int cote)
313 {
314     int ligne; /* numero de ligne, de bas en haut */
315     int colonne; /* numero de colonne, de gauche a droite */
316     for (ligne = cote - 1; ligne >= 0; ligne = ligne - 1) /* pour chaque ligne */
317     {
318         for (colonne = 0; colonne < cote; colonne = colonne + 1) /* pour chaque colonne */
319         {
320             if (motif(colonne, ligne)) /* le point appartient au motif */
321             {
322                 printf("* ");
323             }
324             else /* le point n'appartient pas au motif */
325             {
326                 printf(" ");
327             }
328         }
329         printf("\n"); /* ligne suivante */
330     }
331 }
332
333 int motif(int colonne, int ligne)
334 {
335     return (0 == (colonne + ligne) % 2)
336         || (colonne < 5);
337 }

```

-
5. Ajouter une entrée dans le menu pour la simulation d'une population de lapins.
 6. Comme vous l'aviez remarqué, la fonction `choix_utilisateur()` échoue à redemander un nombre si l'utilisateur saisit autre chose que des chiffres. Vous pouvez corriger ça en utilisant le code suivant à la place de `scanf("%d", &choix)`.

```

char saisie[256]; /* declaration d'un chaine qui nous servira de
                  * "tampon" */

```

```

fgets(saisie, 256, stdin); /* preleve la ligne tapee par
                           l'utilisateur et la place dans le
                           tampon */
sscanf(saisie, "%d", &choix); /* tente de reconnaitre un entier
                              dans le tampon */

```

7. Ajouter au menu une entrée `calcullette` et écrire une procédure `calcullette()` qui affichera le résultat d'une expression `nombre opération nombre` entrée par l'utilisateur, où les nombres sont des `double` et l'opération un caractère parmi `+`, `-`, `*`, `/`. Indication : vous pouvez vous inspirer du code suivant :

```

...
scanf("%lg %c %lg", &x, &op, &y);
if ('+' == op) /* faire une addition */
{
    printf("%lg\n", x + y); /* affichage du résultat */
}

```