
Travaux dirigés 10 : écriture et appel de fonctions et procédures (2)

1 Trace de fonctions

1. Faire la trace du programme suivant.

```
1  /* Declaration de fonctionnalites supplementaires */
2  #include <stdlib.h> /* EXIT_SUCCESS */
3  #include <stdio.h> /* printf() */
4
5  /* Declarations constantes et types utilisateurs */
6
7  /* Declarations de fonctions utilisateurs */
8  void permute_valeurs(int a,int b);
9
10 /* Fonction principale */
11 int main()
12 {
13     /* Declaration et initialisation des variables */
14     int x = 1;
15     int y = 2;
16
17     permute_valeurs(x,y);
18     printf("x = %d et y = %d\n",x,y);
19
20     /* Valeur fonction */
21     return EXIT_SUCCESS;
22 }
23
24 /* Definitions de fonctions utilisateurs */
25 void permute_valeurs(int a,int b)
26 {
27     int aux; /*var aux pour permutation */
28
29     aux = a;
30     a = b;
31     b = aux;
32 }
```

Correction. L'échange n'a pas lieu à cause du mécanisme d'appel de fonction.

main()			
ligne	x	y	Affichage (sortie écran)
initialisation	1	2	
18			x = 1 et y = 2
21			SORTIE AVEC SUCCÈS

permuter_valeurs(1, 2)				
ligne	a	b	aux	Affichage
initialisation	1	2	?	
29			1	
30	2			
31		1		
32	ne renvoie rien			

TAB. 1 – Trace du programme de l'exercice 1.

2 Le menu avec fonctions et procédures

Dans cet exercice vous complétez le menu écrit en TP. Vous travaillerez sur trois parties du programme :

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

2. Déclarer et définir une procédure qui affichera le menu.

Correction. C'est tout bête.

```

/* Declarations 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) Calculette *\n");
    printf("* 3) Deviner un nombre *\n");
    printf("* 4) Motif d'etoiles *\n");
    printf("*\n");
    printf("* 0) QUITTER *\n");
    printf("*\n");
    printf("***** votre choix :");
}

```

3. Déclarer et définir une fonction `choix_utilisateur`, sans paramètres qui renverra une valeur entière saisie par l'utilisateur.

Correction.

```

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

```

4. Déclarer et définir une fonction `executer_menu` qui :
 - affichera le menu à l'utilisateur et réalisera la saisie de son choix;
 - lorsque ce choix est 1, appellera une fonction non encore définie `menu_premier`;

- 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;
}
```

5. Déclarer et définir une procédure `menu_premier` qui traitera le choix 1, et qui fera appel à la fonction `est_premier` du TP (dont vous appellerez la définition et la déclaration) 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;
}

```

6. Écrire le main de telle sorte qu'il 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;
}

```

7. Faire la trace de votre programme dans le cas où l'utilisateur saisit 1 puis 3 puis 0.

Correction. Avant de faire la trace il nous faut des numéros de ligne :

```

1  /* Fonctionnalités supplémentaires */
2  #include <stdlib.h> /* EXIT_SUCCESS */
3  #include <stdio.h> /* printf(), scanf() */
4
5  /* Declarations de types et constantes utilisateurs */
6  #define TRUE 1
7  #define FALSE 0
8
9  /* Declarations de fonctions utilisateurs */
10
11 /* Affiche le menu */
12 void afficher_menu();
13
14 /* Recupere un nombre entier saisi par l'utilisateur et le retourne */
15 int choix_utilisateur();
16
17 /* Affichage du menu et traitement du choix de l'utilisateur */
18 int executer_menu();
19
20 /* Teste si un nombre saisi par l'utilisateur est premier et affiche le resultat */
21 void menu_premier();
22
23 /* Teste si son argument (positif) est premier */
24 int est_premier(int p);
25
26
27 /* Fonction principale */
28 int main()
29 {

```

```

30     /* Declarations et initialisation des variables */
31     int encore = TRUE;
32
33     /* Boucle d'interaction avec l'utilisateur */
34     while (encore)
35     {
36         encore = executer_menu();
37     }
38
39     /* Greetings */
40     printf("Bye bye\n");
41
42     /* Valeur fonction */
43     return EXIT_SUCCESS;
44 }
45
46
47
48 /* Definitions de fonctions utilisateurs */
49 void afficher_menu()
50 {
51     printf("***** MENU *****\n");
52     printf("*\n");
53     printf("* 1) Tester si un nombre est premier *\n");
54     printf("* 2) Calculette *\n");
55     printf("* 3) Deviner un nombre *\n");
56     printf("* 4) Motif d'etoiles *\n");
57     printf("*\n");
58     printf("* 0) QUITTER *\n");
59     printf("*\n");
60     printf("***** votre choix :");
61 }
62
63 int choix_utilisateur()
64 {
65     int choix;
66     scanf("%d", &choix);
67     return choix;
68 }
69
70 int executer_menu()
71 {
72     int choix;
73
74     /* Affichage du menu et choix de l'utilisateur */
75     afficher_menu();
76     choix = choix_utilisateur();
77
78     if (1 == choix) /* ----- 1) Tester si un nombre est premier ----- */
79     {
80         menu_premier();
81     }
82
83     /* Valeur de retour */
84     if (choix != 0)
85     {
86         return TRUE;
87     }
88     return FALSE;
89 }
90
91 void menu_premier()

```

```

92  {
93      int p;
94
95      printf("Donner un nombre : ");
96      p = choix_utilisateur();
97
98      if (est_premier(p))
99      {
100         printf("Le nombre %d est premier\n", p);
101     }
102     else
103     {
104         printf("Le nombre %d n'est pas premier\n", p);
105     }
106 }
107
108
109 int est_premier(int n)
110 {
111     int i = 2;
112     int premier = TRUE;
113
114     while (premier && i < n)
115     {
116         if (n % i == 0)
117         {
118             premier = FALSE;
119         }
120         i = i + 1;
121     }
122     return premier;
123 }

```

On peut maintenant faire la trace (Table ?? page ??). La trace est donnée ici avec l'arbre d'appel (avec partage). Inutile de tout faire, le second appel à `executer_menu` est notamment un peu superflu.

8. 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 soit affiché à l'utilisateur et la saisie redemandée, jusqu'à cinq fois.

Correction. Au delà de cinq essais, c'est la dernière saisie utilisateur qui est renvoyée.

```

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;
    }
    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 `INT_MAX`).

On se donne la procédure suivante :

```
void affichage_motif(int cote)
{
    int ligne; /* numero de ligne, de bas en haut */
    int colonne; /* numero de colonne, de gauche a droite */
    for (ligne = cote - 1; ligne >= 0; ligne = ligne - 1) /* pour chaque ligne */
    {
        for (colonne = 0; colonne < cote; colonne = colonne + 1) /* pour chaque colonne */
        {
            if (motif(colonne, ligne)) /* le point appartient au motif */
            {
                printf("* ");
            }
            else /* le point n'appartient pas au motif */
            {
                printf(" ");
            }
        }
        printf("\n"); /* ligne suivante */
    }
}
```

9. Définir la fonction motif de telle sorte qu'un appel à `affichage_motif(5)` affiche :

```
*      *
*     *
*    *
*   *
*  *
* * * * *
```

Correction. Facile :

```
int motif(int x, int y)
{
    return x == y
        || x == 0
        || y == 0;
}
```

10. Écrire les fonctions nécessaires au traitement du choix 3 du menu sur le modèle du traitement du choix 1.

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).

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

```
1  /* Fonctionnalites supplementaires */
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  /* Declarations 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 /* Declarations de fonctions utilisateurs */
14
15 /* Affiche le menu */
```

```

16 void afficher_menu();
17
18 /* Recupere 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 resultat */
26 void menu_premier();
27
28 /* Teste si son argument (positif) est premier */
29 int est_premier(int p);
30
31 /* Jouer a une devinette */
32 void menu_deviner();
33
34 /* Tirer un nombre au hasard entre 0 et n */
35 int nombre_aleatoire(int n);
36
37 /* Fonction principale */
38 int main()
39 {
40     /* Declarations et initialisation des variables */
41     int encore = TRUE;
42
43     /* Initialisation du generateur aleatoire */
44     srand(time(NULL)); /* à ne faire qu'une fois */
45
46     /* Boucle d'interaction avec l'utilisateur */
47     while (encore)
48     {
49         encore = executer_menu();
50     }
51
52     /* Greetings */
53     printf("Bye bye\n");
54
55     /* Valeur fonction */
56     return EXIT_SUCCESS;
57 }
58
59
60
61 /* Definitions de fonctions utilisateurs */
62 void afficher_menu()
63 {
64     printf("***** MENU *****\n");
65     printf("*\n");
66     printf("* 1) Tester si un nombre est premier *\n");
67     printf("* 2) Calculette *\n");
68     printf("* 3) Deviner un nombre *\n");
69     printf("* 4) Motif d'etoiles *\n");
70     printf("*\n");
71     printf("* 0) QUITTER *\n");
72     printf("*\n");
73     printf("***** votre choix : ");
74 }
75
76 int choix_utilisateur(int a, int b)
77 {

```

```

78     int compteur = 5; /* compteur du nombre d'essais */
79     int choix; /* choix de l'utilisateur */
80
81     scanf("%d", &choix);
82     while ((compteur > 0) && ((choix < a) || (choix > b)))
83     {
84         printf("Le nombre doit etre entre %d et %d (inclus) : ", a, b);
85         scanf("%d", &choix);
86         compteur = compteur - 1;
87     }
88     return choix;
89 }
90
91 int executer_menu()
92 {
93     int choix;
94
95     /* Affichage du menu et choix de l'utilisateur */
96     afficher_menu();
97     choix = choix_utilisateur(0, 4);
98
99     if (1 == choix) /* ----- 1) Tester si un nombre est premier ----- */
100    {
101        menu_premier();
102    }
103
104     if (3 == choix) /* ----- 3) Deviner un nombre ----- */
105    {
106        menu_deviner();
107    }
108
109
110     /* Valeur de retour */
111     if (choix != 0)
112     {
113         return TRUE;
114     }
115     return FALSE;
116 }
117
118 void menu_premier()
119 {
120     int p;
121
122     printf("Donner un nombre : ");
123     p = choix_utilisateur(1, INT_MAX);
124
125     if (est_premier(p))
126     {
127         printf("Le nombre %d est premier\n", p);
128     }
129     else
130     {
131         printf("Le nombre %d n'est pas premier\n", p);
132     }
133 }
134
135
136 int est_premier(int n)
137 {
138     int i = 2;
139     int premier = TRUE;

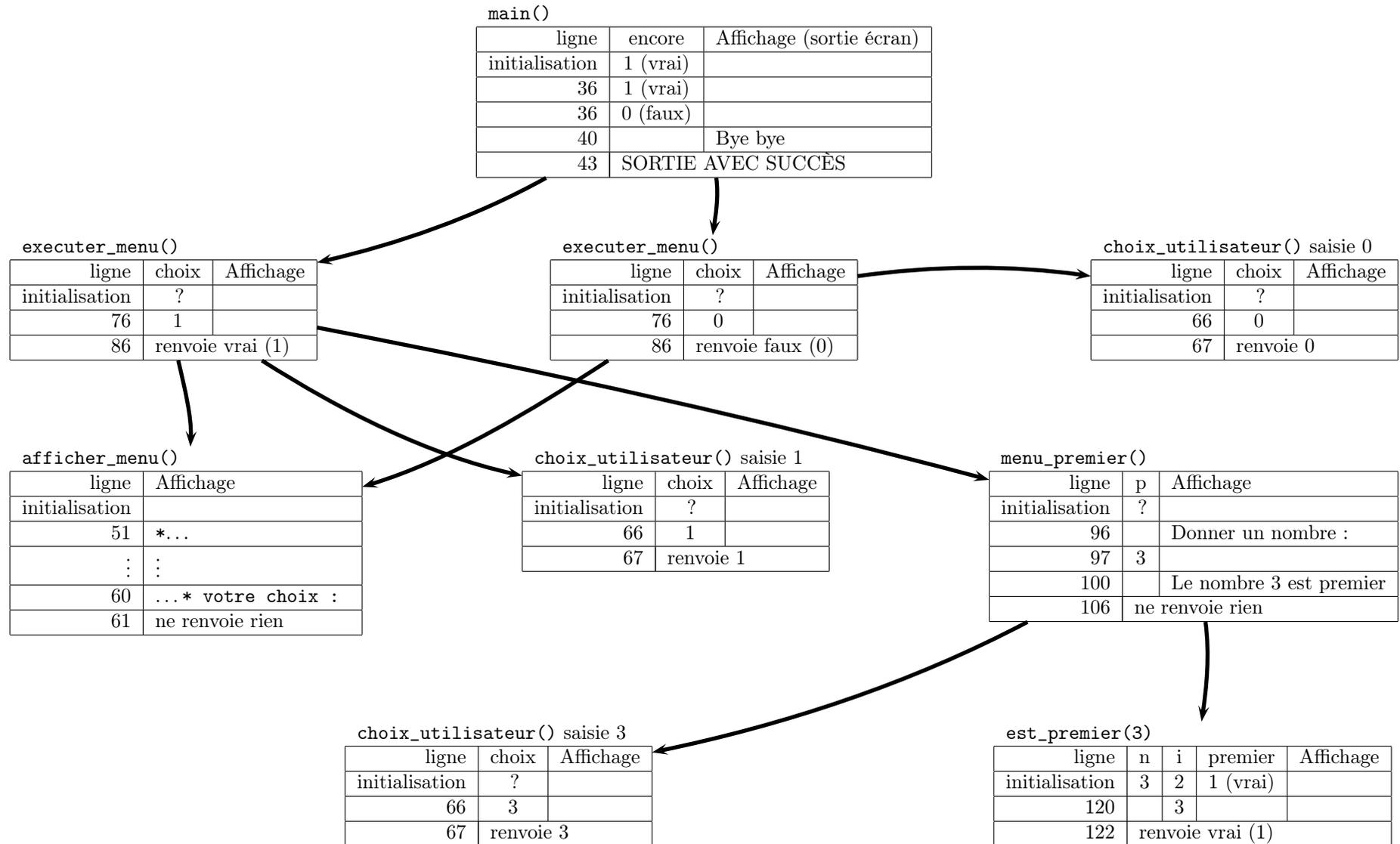
```

```

140
141     while (premier && i < n)
142     {
143         if (n % i == 0)
144         {
145             premier = FALSE;
146         }
147         i = i + 1;
148     }
149     return premier;
150 }
151
152
153 void menu_deviner()
154 {
155     int choix; /* choix de l'utilisateur pour le nombre secret */
156     int trouve = FALSE; /* TRUE si trouvé */
157     int nombre_secret;
158     int essais = DEVINER_ESSAIS; /* essais restants */
159
160     /* Tirage aléatoire du nombre secret */
161     nombre_secret = nombre_aleatoire(DEVINER_MAX);
162
163     /* manche joueur */
164     while(!trouve && (essais > 0)) /* pas trouvé nombre secret */
165     {
166         /* demande nombre à l'utilisateur */
167         printf("Votre choix (nombre entre 0 et %d) : ", DEVINER_MAX);
168         choix = choix_utilisateur(0, DEVINER_MAX);
169
170         if(choix == nombre_secret) /* trouvé */
171         {
172             trouve = TRUE;
173         }
174         else /* pas trouvé */
175         {
176             /* donne indice */
177             if(choix > nombre_secret)
178             {
179                 printf("Trop grand.\n");
180             }
181             else
182             {
183                 printf("Trop petit.\n");
184             }
185         }
186         essais = essais - 1;
187     }
188
189     if (essais > 0)
190     {
191         /* trouvé nombre secret */
192         printf("Gagné. Vous avez trouvé le nombre secret.\n");
193     }
194     else
195     {
196         /* Perdu */
197         printf("Perdu : limite du nombre d'essais atteinte.\n");
198     }
199 }
200
201

```

```
202 int nombre_aleatoire(int n)
203 {
204     /* tirage du nombre secret */
205     return rand() % (n + 1); /* entre 0 et n inclus */
206 }
```



TAB. 2 – Trace du programme de l'exercice 2.