

ECUE « Introduction à la programmation » - Session 2
14 juin 2017 - Bruno Bouzy
sans document - durée 1 heure 30

CORRIGE

Exercice 1 (3 points)

Ecrire un programme `ex01_vote.c` permettant à l'utilisateur d'entrer un nombre de votes « pour » et un nombre de votes « contre » et affichant le vainqueur éventuel avec le pourcentage de succès. La sortie du programme doit respecter la sortie ci-dessous. On suppose que l'utilisateur entre des valeurs strictement positives. On affichera le pourcentage avec un chiffre après la virgule. Les caractères gras sont en entrée du programme. En cas d'égalité le programme affiche `EGALITE`.

```
nombre de votes POUR ? 200
nombre de votes POUR = 200
nombre de votes CONTRE ? 125
nombre de votes CONTRE = 125
vainqueur : POUR a 61.5
```

```
#include <stdio.h>
int main() {
    float pour, contre;
    printf("nombre de votes POUR ? ");
    scanf("%f", &pour);
    printf("nombre de votes POUR = %.0f\n", pour);
    printf("nombre de votes CONTRE ? ");
    scanf("%f", &contre);
    printf("nombre de votes CONTRE = %.0f\n", contre);
    printf("vainqueur: ");
    if (pour>contre)    printf("POUR");
    else if (pour==contre) printf("EGALITE");
    else                printf("CONTRE");
    printf(" a %.1f\n", 100*pour/(pour+contre));
    return 0;
}
```

Exercice 2 (4 points)

Donner la sortie du programme ci-dessous. Pour chaque ligne en **caractères gras**, tenir compte de la **couleur de votre copie (bleu, rouge, vert, jaune)** pour **valoriser a et b** avec les valeurs précisées dans le commentaire de la ligne.

```
#include <stdio.h>
int main() {
    int a = 3; // Couleur de la copie : Bleu:3, Rouge:4, Vert:5, Jaune:7
    int * p = &a;
    int b = *p; printf("1: a = %d, b = %d, *p = %d.\n", a, b, *p);
    a += 4; // Couleur de la copie : Bleu:4, Rouge:5, Vert:7, Jaune:3
    printf("2: a = %d, b = %d, *p = %d.\n", a, b, *p);
    b *= 5; // Couleur de la copie : Bleu:5, Rouge:7, Vert:3, Jaune:4
    printf("3: a = %d, b = %d, *p = %d.\n", a, b, *p);
    int * q = &b; printf("4: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    *q *= (*p)++; printf("5: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    *q -= ++(*p); printf("6: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    p = q; printf("7: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    q = &a; printf("8: a = %d, b = %d, *p = %d, *q = %d.\n", a, b, *p, *q);
    return 0;
}
```

Copies bleues :

```
1: a = 3, b = 3, *p = 3.
2: a = 7, b = 3, *p = 7.
3: a = 7, b = 15, *p = 7.
4: a = 7, b = 15, *p = 7, *q = 15.
5: a = 8, b = 105, *p = 8, *q = 105.
6: a = 9, b = 96, *p = 9, *q = 96.
7: a = 9, b = 96, *p = 96, *q = 96.
8: a = 9, b = 96, *p = 96, *q = 9.
```

Copies rouges :

```
1: a = 4, b = 4, *p = 4.
2: a = 9, b = 4, *p = 9.
3: a = 9, b = 28, *p = 9.
4: a = 9, b = 28, *p = 9, *q = 28.
5: a = 10, b = 252, *p = 10, *q = 252.
6: a = 11, b = 241, *p = 11, *q = 241.
7: a = 11, b = 241, *p = 241, *q = 241.
8: a = 11, b = 241, *p = 241, *q = 11.
```

Copies vertes :

```
1: a = 5, b = 5, *p = 5.
2: a = 12, b = 5, *p = 12.
3: a = 12, b = 15, *p = 12.
4: a = 12, b = 15, *p = 12, *q = 15.
5: a = 13, b = 180, *p = 13, *q = 180.
6: a = 14, b = 166, *p = 14, *q = 166.
7: a = 14, b = 166, *p = 166, *q = 166.
8: a = 14, b = 166, *p = 166, *q = 14.
```

Copies jaunes :

```
1: a = 7, b = 7, *p = 7.
2: a = 10, b = 7, *p = 10.
3: a = 10, b = 28, *p = 10.
4: a = 10, b = 28, *p = 10, *q = 28.
5: a = 11, b = 280, *p = 11, *q = 280.
6: a = 12, b = 268, *p = 12, *q = 268.
7: a = 12, b = 268, *p = 268, *q = 268.
8: a = 12, b = 268, *p = 268, *q = 12.
```

Exercice 3 (3 points)

Pour chaque question, donner la sortie du programme.

1)

```
void cdt1(int t) { if (t==1) ; printf("A ");           printf("B "); }
void cdt2(int t) { if (t==1)  printf("A ");           printf("B "); }
void cdt3(int t) { if (t==1) { printf("A "); return; } printf("B "); }
int main() {
    printf("0: ");  cdt1(0); cdt2(0); cdt3(0);
    printf("\n1: "); cdt1(1); cdt2(1); cdt3(1); printf("\n"); return 0;
}
```

```
0: A B B B
1: A B A B A
```

2)

```
void whl1(int t) { int i=t; while (i++<=3); printf("V "); printf("W "); }
void whl2(int t) { int i=t; while (i++<=3)  printf("V "); printf("W "); }
void whl3(int t) { int i=t; while (i++<=3) { printf("V "); printf("W "); } }
int main() {
    printf("1: ");  whl1(1); printf("\n2: "); whl2(1);
    printf("\n3: "); whl3(1); printf("\n"); return 0;
}
```

```
1: V W
2: V V V W
3: V W V W V W
```

3)

```
void for1(int t) { int i=0; for (i=t; i<=3; i++) printf("F "); printf("G "); }
void for2(int t) { int i=0; for (  ; i<=3; i++) printf("F "); printf("G "); }
void for3(int t) { int i=0; for (i=t;      ; i++) printf("F "); printf("G "); }
int main() {
    printf("1: ");  for1(1); printf("\n2: "); for2(1);
    printf("\n3: "); for3(1); printf("\n"); return 0;
}
```

```
1: F F F G
2: F F F F G
3: F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F F
F F F F F ...
```

Exercice 4 (10 points)

1) Ecrire une fonction `int deIntervalleANombre(int a, int b)` demandant à l'utilisateur un nombre entier appartenant à l'intervalle $[a, b]$ et retournant ce nombre. Tant que le nombre n'appartient pas à $[a, b]$, la fonction re-demande le nombre à l'utilisateur **(2 points)**.

```
int deIntervalleANombre(int a, int b)
{
    int x;
    do {
        printf (" x ? (%d<=x<=%d) ", a, b);
        scanf("%d", &x);
    } while (x>b||x<a);
    return x;
}
```

2) Ecrire une fonction `void deTabANombresClavier` prenant en entrée deux entiers `a` et `b`, un tableau d'entiers `tab` et une longueur `l` de tableau, et remplissant le tableau `tab` avec des nombres distincts appartenant à l'intervalle $[a, b]$ avec des appels à la fonction `deIntervalleANombre(int a, int b)` ci-dessus. **(2 points)**

```
int deTabANombresClavier(int a, int b, int * tab, int l)
{
    for (int i=0; i<l; i++) {
        printf("t[%d] : ", i);
        int h, deja;
        do {
            h = deIntervalleANombre(a, b);
            deja = 0;
            for (int j=0; j<i; j++) if (tab[j]==h) deja = 1;
        } while (deja==1);
        tab[i] = h;
    }
}
```

3) Ecrire une fonction `void deTabANombresAleatoires` prenant en entrée deux entiers `a` et `b`, un tableau d'entiers `tab` et une longueur `l` de tableau, et remplissant aléatoirement le tableau `tab` avec des nombres distincts appartenant à l'intervalle $[a, b]$, avec des appels appropriés à la fonction `rand()`. **(2 points)**

```
int deTabANombresAleatoires(int a, int b, int * tab, int l)
{
    for (int i=0; i<l; i++) {
        int h, deja;
        do {
            h = a + rand() % (1+b-a);
            deja = 0;
            for (int j=0; j<i; j++) if (tab[j]==h) deja = 1;
        } while (deja==1);
        tab[i] = h;
    }
}
```

4) Ecrire un programme main simulant le jeu Euromillions respectant l'énoncé suivant.

En utilisant un appel à la fonction `deTabANombresClavier`, le programme Euromillions demande à l'utilisateur (le joueur) de remplir un tableau de 5 numéros appartenant à l'intervalle $[1, 50]$ **(0.5 point)**.

Puis le programme affiche les numéros du joueur à l'écran **(0.5 point)**.

En utilisant un appel à la fonction `deTabANombresAleatoires`, le programme Euromillions remplit un tableau de 5 numéros aléatoires appartenant à l'intervalle $[1, 50]$ **(0.5 point)**.

Puis le programme affiche les numéros gagnants à l'écran **(0.5 point)**.

Enfin, le programme compare les numéros du joueur avec ceux du tirage gagnant : il affiche les numéros corrects et leur nombre **(1 point)**.

On veillera à la correction de l'ensemble du programme (constantes, déclarations, variables, tableaux, fonctions, affichages) **(1 point)**.

Pour la fonction `deIntervalleANombre`, on respectera les entrée-sorties suivantes:

```
t[0] : x ? (1<=x<=50) 53
x ? (1<=x<=50) 24
t[1] : x ? (1<=x<=50) 9
t[2] : x ? (1<=x<=50) 5
t[2] : x ? (1<=x<=50) 47
t[2] : x ? (1<=x<=50) 38
t[0] = 24, t[1] = 9, t[2] = 5, t[3] = 47, t[4] = 38.
```

```
// exo4_euromillions.c

#include <stdio.h>
#include <stdlib.h>
#include <time.h>

#define NUMERO_MAX 50
#define NOMBRE_NUMEROS 7

// ici les définitions des fonctions.

int main() {

    int joueur[NOMBRE_NUMEROS];
    int tirage[NOMBRE_NUMEROS];

    // 1 point
    printf("Numéros du joueur:\n");
    deTabANombresClavier(1, NUMERO_MAX, joueur, NOMBRE_NUMEROS);
    for (int i=0; i<NOMBRE_NUMEROS; i++) printf( "%d ", joueur[i]);
    printf("\n");

    // 1 point
    srand(time(NULL));
    printf("Tirage gagnant: ");
    deTabANombresAleatoires(1, NUMERO_MAX, tirage, NOMBRE_NUMEROS);
    for (int i=0; i<NOMBRE_NUMEROS; i++) printf( "%d ", tirage[i]);
    printf("\n");

    // 1 point
    printf("Numéros gagnants du joueur: ");
    int nng=0;
    for (int i=0; i<NOMBRE_NUMEROS; i++) {
        for (int j=0; j<NOMBRE_NUMEROS; j++) {
            if (joueur[i]==tirage[j]) {
                printf("%d ", joueur[i]);
                nng++;
            }
        }
    }
    printf("\n");
    printf("Nombre de numéros gagnants = %d\n", nng);

    return 0;
}
```