

## ECUE «Introduction à la programmation »

Contrôle continu n°1 (CC1)  
31 octobre 2016  
sans document - durée 1 heure 30

Dans tous les exercices, les entrées clavier sont indiquées en caractères gras.

Corrigé

Corrigé

Corrigé

Corrigé

**Exercice 1 (4 points)**

Corriger le programme suivant (8 erreurs à trouver).

```
$include <stdio.h>
int main{} {
    int m, n;
    printf("m ? "):
    scanf("%d", &m);
    printf("m = %d\n", m);
    printf("n ? ");
    scanf("%d", n);
    printf("n = %d\n", n);
    printf("m+n = %d\n", sum);
    printf("m-n = %d\n", m-n);
    retrun 0;
}
```

// 8 erreurs.

```
$include <stdio.h> // $ -> #
int main{} { // {} -> ()
    int m, n;
    printf("m ? ") : // : -> ;
    scanf("%d", &m);
    printf("m = %d\n", m); // ," -> ",
    printf("n ? ");
    scanf("%d", n); // n -> &n
    printf("n = %d\n", n);
    printf("m+n = %d\n", sum); // sum -> m+n
    printf("m-n = %d\n", m-n); // \m -> \n
    retrun 0; // retrun -> return
}
```

**0.5 point par erreur trouvée**

## Exercice 2 (5 points)

Ecrire un programme `exo2.c` permettant à l'utilisateur d'entrer 3 nombres entiers  $a, b, c$  et les affichant en ordre croissant, séparés par un  $<$  ou un  $=$  selon qu'ils sont différents ou égaux deux à deux. La sortie du programme doit correspondre aux exécutions ci-dessous.

a ? 1	a ? 1	a ? 3	a ? 3	a ? 2
b ? 2	b ? 1	b ? 2	b ? 2	b ? 2
c ? 3	c ? 2	c ? 1	c ? 3	c ? 2
1<2<3	1=1<2	1<2<3	2<3=3	2=2=2

```
// exo2.c
#include <stdio.h>
int main() {
    int a, b, c, d; // 0.5 point pour le début

    printf("a ? "); scanf ("%d", &a); printf("a = %d\n", a);
    printf("b ? "); scanf ("%d", &b); printf("b = %d\n", b);
    printf("c ? "); scanf ("%d", &c); printf("c = %d\n", c);
    // 0.5 point pour les printf scanf
    if (a>b) { d=a; a=b; b=d; }
    if (b>c) { d=b; b=c; c=d; }
    if (a>b) { d=a; a=b; b=d; }
    // 2 points pour les if corrects
    printf("%d", a);
    if (a==b) printf("=");
    else printf("<");
    printf("%d", b);
    if (b==c) printf("=");
    else printf("<");
    printf("%d\n", c);
    // 2 points pour les if corrects
    return (0);
}
```

## Exercice 3 (7 points)

Ecrire un programme `hanabi.c` permettant à l'utilisateur d'entrer les hauteurs de trois feux d'artifices puis une carte (hauteur et couleur) et vérifiant que la carte entrée est compatible avec une hauteur de feu d'artifices. Chaque feu a une couleur : un feu rouge, un feu bleu et un feu vert. Les hauteurs de feux sont comprises entre 0 et 3 inclus. Les hauteurs de cartes sont comprises entre 1 et 3 inclus. Les couleurs des cartes sont représentées par des constantes : 1 pour rouge, 2 pour bleu et 3 pour vert. Si une hauteur de feu entrée par l'utilisateur n'est pas dans  $\{0, 1, 2, 3\}$ , le programme affiche une erreur et retourne -1. Sinon il affiche la hauteur des trois feux et continue. Si la hauteur ou la couleur de la carte entrée n'est pas dans  $\{1, 2, 3\}$ , le programme affiche une erreur et retourne -1. Sinon il teste si la hauteur de la carte est égale à la hauteur du feu de la

couleur de la carte plus un. Si oui, il met à jour la hauteur des feux, les affiche et retourne +1. Sinon, il affiche une erreur et retourne -1. La sortie du programme doit correspondre aux exécutions ci-dessous. On utilisera des constantes ROUGE, BLEU, VERT pour les couleurs dans {1, 2, 3}. On utilisera les variables suivantes : hrouge, hbleu, hvert, ccarte, hcarte.

```
FEUX D'ARTIFICES...
Hauteur feu rouge ? 4
erreur: hauteur incorrecte.
```

```
FEUX D'ARTIFICES...
Hauteur feu rouge ? 1
Hauteur feu bleu ? -1
erreur: hauteur incorrecte.
```

```
FEUX D'ARTIFICES...
Hauteur feu rouge ? 1
Hauteur feu bleu ? 2
Hauteur feu vert ? 3
Rouge: 1, Bleu: 2, Vert: 3 .
CARTE JOUEE...
Couleur (1 Rouge) (2 Bleu) (3 Vert)? 2
Hauteur? (1, 2 ou 3) 2
La carte jouee est bleue de hauteur 2.
erreur: 2 different de 2+1.
```

```
FEUX D'ARTIFICES...
Hauteur feu rouge ? 1
Hauteur feu bleu ? 2
Hauteur feu vert ? 3
Rouge: 1, Bleu: 2, Vert: 3 .
CARTE JOUEE...
Couleur (1 Rouge) (2 Bleu) (3 Vert)? 0
erreur: couleur incorrecte.
```

```
FEUX D'ARTIFICES...
Hauteur feu rouge ? 1
Hauteur feu bleu ? 2
Hauteur feu vert ? 3
Rouge: 1, Bleu: 2, Vert: 3 .
CARTE JOUEE...
Couleur (1 Rouge) (2 Bleu) (3 Vert)? 1
Hauteur (1, 2 ou 3)? 2
La carte jouee est rouge de hauteur 2.
BRAVO !!!
Rouge: 2, Bleu: 2, Vert: 3 .
```

```
// hanabi.c
#include <stdio.h>

#define ROUGE 1
#define BLEU 2
#define VERT 3

int main() {
    int hrouge, hbleu, hvert;
    int ccarte, hcarte; // 0.5 point pour ce qui précède

    printf("FEUX D'ARTIFICES...\n");
    printf("Hauteur feu rouge ? "); scanf("%d", &hrouge);
    if ((hrouge<0) || (hrouge>3)) {
        printf("erreur: hauteur incorrecte.\n");
        return -1;
    }
    printf("Hauteur feu bleu ? "); scanf("%d", &hbleu);
    if ((hbleu<0) || (hbleu>3)) {
        printf("erreur: hauteur incorrecte.\n");
        return -1;
    }
    printf("Hauteur feu vert ? "); scanf("%d", &hvert);
    if ((hvert<0) || (hvert>3)) {
        printf("erreur: hauteur incorrecte.\n");
        return -1;
    }

    printf("Rouge: %d, Bleu: %d, Vert: %d.\n", hrouge, hbleu, hvert);
    // 2 points pour les feux
```

```

printf("CARTE JOUEE...\n");
printf("Couleur (%d Rouge) (%d Bleu) (%d Vert) ? ", ROUGE, BLEU, VERT);
scanf("%d", &ccarte);
if ((ccarte<ROUGE) || (ccarte>VERT)) {
    printf("erreur: couleur incorrecte.\n"); return -1; }
printf("Hauteur (1, 2 ou 3) ? ");
scanf("%d", &hcarte);
if ((hcarte<1) || (hcarte>3)) {
    printf("erreur: hauteur incorrecte.\n"); return -1; }
// 2 points pour la carte

if (ccarte==ROUGE) {
    printf("La carte jouee est rouge de hauteur %d\n", hcarte);
    if (hcarte==hrouge+1) hrouge++;
    else {
        printf("erreur: %d different de %d+1.\n", hcarte, hrouge); return -1; }
    }
else if (ccarte==BLEU) {
    printf("La carte jouee est bleue de hauteur %d\n", hcarte);
    if (hcarte==hbleu+1) hbleu++;
    else {
        printf("erreur: %d different de %d+1.\n", hcarte, hbleu); return -1; }
    }
else if (ccarte==VERT) {
    printf("La carte jouee est verte de hauteur %d\n", hcarte);
    if (hcarte==hvert+1) hvert++;
    else {
        printf("erreur: %d different de %d+1.\n", hcarte, hvert); return -1; }
    } // 2 points pour le traitement ci-avant

printf("BRAVO !!!\n");
printf("Rouge: %d, Bleu: %d, Vert: %d.\n", hrouge, hbleu, hvert);
return +1;
}

// 0.5 point pour l'ensemble

```

## Exercice 4 (4 points)

Donner la sortie du programme exo4.c.

```

// exo4.c
#include <stdio.h>
int main() {
    // enlever le commentaire a gauche de la ligne
    // correspondant a la couleur de votre copie
    // int a=4, b=3, c=2; // copie rouge
    // int a=3, b=2, c=4; // copie bleue
    // int a=2, b=4, c=3; // copie verte
    // int a=4, b=2, c=3; // copie jaune
    printf("a+b*c=%d\n", a+b*c);
    printf("(a+b)*c=%d\n", (a+b)*c);
    printf("c/a = %d reste %d\n", c/a, c%a);
    a+=b+c;    printf("a=%d, b=%d, c=%d\n", a, b, c);
}

```

```
        b--a+c;      printf("a=%d, b=%d, c=%d\n", a, b, c);
        c*=a*b;      printf("a=%d, b=%d, c=%d\n", a, b, c);
        a=a+(b++);   printf("a=%d, b=%d\n", a, b);
        a=a(++b);    printf("a=%d, b=%d\n", a, b);
        return (0);
    }
```

**0.5 point par ligne correcte.**

Si une ligne est fausse, la suite est fausse.

**Rouge :**

```
a+b*c=10
(a+b)*c=14
c/a = 0 reste 2
a=9, b=3, c=2
a=9, b=-8, c=2
a=9, b=-8, c=-144
a=1, b=-7
a=-5, b=-6
```

**Bleu :**

```
a+b*c=11
(a+b)*c=20
c/a = 1 reste 1
a=9, b=2, c=4
a=9, b=-11, c=4
a=9, b=-11, c=-396
a=-2, b=-10
a=-11, b=-9
```

**Vert :**

```
a+b*c=14
(a+b)*c=18
c/a = 1 reste 1
a=9, b=4, c=3
a=9, b=-8, c=3
a=9, b=-8, c=-216
a=1, b=-7
a=-5, b=-6
```

**Jaune :**

```
a+b*c=10
(a+b)*c=18
c/a = 0 reste 3
a=9, b=2, c=3
a=9, b=-10, c=3
a=9, b=-10, c=-270
a=-1, b=-9
a=-9, b=-8
```