

## 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.

**Exercice 1 (4 points)**

Trouver et corriger les 8 erreurs du programme suivant :

```

#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;
}

```

**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 cinq exécutions ci-dessous.

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

**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, de carte ou de feu, sont représentées par des nombres entiers: 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 de ce feu avec la hauteur de la carte, il affiche les feux et retourne +1. Sinon, il affiche une erreur et retourne -1. La sortie du programme doit correspondre aux cinq exécutions ci-dessous.

On utilisera des constantes, ROUGE, BLEU, VERT, pour les trois couleurs dans {1, 2, 3}. On utilisera les cinq variables de type `int` suivantes : `hrouge` (respectivement `hbleu`, `hvert`) pour la hauteur du feu d'artifice rouge (respectivement bleu, vert), `ccarte` (respectivement `hcarte`) pour la couleur (respectivement hauteur) de la carte.

```
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.
```

## Exercice 4 (4 points)

Donner la sortie du programme `exo4.c` ci-dessous, modifié en enlevant le `//` situé à gauche de la ligne de déclaration des variables `a`, `b`, `c`, correspondant à la couleur de votre copie.

```
// exo4.c
#include <stdio.h>
int main() {
    // 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;
}
```