

CS-119(a) – ICC-C Série 3

2024-03-05

Exo1 Expressions tordues

Qu'affichent les instructions suivantes? Essayez de deviner avant de les faire tourner. Trouvez une explication en vous souvenant des conversions implicites et de l'ordre des opérations.

```
1 printf("Expression A: %d\n", 0.01 == 0.01 == (-3 && "test"));
2 printf("Expression B: %d\n", 4 && 0 || 10 + 3);
3 int a = 0;
4 printf("Expression C: %d\n", a++ || 7 - 7 || !5);
```

Solution de l'exercice 1

Les valeurs `-3` et `"test"` ne sont pas `0`, donc par conversion booléenne elles sont converties à `1`. Ensuite il faut utiliser le fait que l'opérateur d'égalité est associatif de gauche à droite : $(0.01 == 0.01) == (1 \ \&\& \ 1)$. Nous obtenons donc `1 == 1`, et donc `1`.

Les opérations arithmétiques sont évaluées avant les opérations booléennes, puis l'opérateur ET a la priorité sur le OU.

```
1 4 && 0 || 10 + 3
2 (4 && 0) || 13
3 0 || 1
4 1
```

L'expression `a++` vaut `0`, quoique la variable `a` vaut `1` après l'incréméntation.

```
1 a++ || 7 - 7 || !5
2 0 || 0 || (!5)
3 0 || 0
4 0
```

Exo2 $ax^2 + bx + c = 0$

Chouette, on peut utiliser le langage C pour résoudre des équations de 2e degré!

Lisez depuis l'entrée standard trois réels a , b , et c , et calculez les deux solutions réelles de l'équation $ax^2 + bx + c = 0$.

Si l'équation admet deux solutions réelles, disons 3.1 et -7, affichez

2 solutions: 3.1 -7

sur la même ligne. Si l'équation n'a qu'une solution, disons 10.5, affichez

1 solution: 10.5

Si l'équation n'admet pas de solution réelle, affichez

Pas de solution

Rappel : les fonctions mathématiques se trouvent dans `<math.h>`. Vous aurez sûrement besoin de la fonction qui calcule la racine carée `sqrt` (square root).

Solution de l'exercice 2

```
1 #include <stdio.h>
2 #include <math.h>
3
4 int main() {
5     double eps = 1e-6;
6     double a, b, c;
7     double discriminant, racine1, racine2;
8
9     // Lire les coefficients de l'équation quadratique
10    printf("Entrez les coefficients (a, b, c) de l'équation
11    quadratique : ");
12    scanf("%lf %lf %lf", &a, &b, &c);
13
14    // Calculer le discriminant
15    discriminant = b * b - 4 * a * c;
16
17    // Vérifier si l'équation a des racines réelles
18    if (discriminant >= eps) {
19        // Calculer les racines
20        racine1 = (-b + sqrt(discriminant)) / (2 * a);
21        racine2 = (-b - sqrt(discriminant)) / (2 * a);
22
23        // Afficher les racines
24        printf("2 solutions: %.2lf %.2lf\n", racine1, racine2);
```

```

24 } else if (fabs(discriminant) < eps) {
25     // Calculer la racine
26     racine1 = -b / (2 * a);
27     printf("1 solution: %.2lf\n", racine1);
28 } else {
29     // Afficher un message d'erreur
30     printf("Pas de solution\n");
31 }
32
33 return 0;
34 }

```

Exo3 Tri de trois nombres

On lit depuis l'entrée standard trois nombres entiers séparés par des espaces. Ce programme va afficher les trois nombres triés par ordre décroissant. Par exemple, si on rentre 3 12 5, le programme devrait afficher 12 5 3.

Solution de l'exercice 3

```

1 #include <stdio.h>
2
3 int main() {
4     int a, b, c;
5
6     printf("Entrez trois entiers : ");
7     scanf("%d %d %d", &a, &b, &c);
8
9     // Tri décroissant
10    if (a >= b && b >= c) {
11        printf("%d %d %d\n", a, b, c);
12    } else if (a >= c && c >= b) {
13        printf("%d %d %d\n", a, c, b);
14    } else if (b >= a && a >= c) {
15        printf("%d %d %d\n", b, a, c);
16    } else if (b >= c && c >= a) {
17        printf("%d %d %d\n", b, c, a);
18    } else if (c >= a && a >= b) {
19        printf("%d %d %d\n", c, a, b);
20    } else {
21        printf("%d %d %d\n", c, b, a);
22    }
23
24    return 0;
25 }

```

Exo4 La journée de Bob

Lisez deux entiers depuis l'entrée standard qui représentent l'heure et les minutes d'un moment pendant la journée de Bob. Par exemple 13 20 veut dire 13h20. On aimerait savoir ce que faisait Bob à ce moment là. Voici ce qu'on sait :

```
minuit - 8h30 Bob dort
8h30 - 9h Bob prend le petit déjeuner
9h - 11h45 Bob travaille
11h45 - 13h Bob mange son repas de midi
13h - 13h30 Bob boit son café
13h30 - 17h Bob travaille
17h - 18h Bob est au pub (happy hour!)
18h - 22h Bob fait sa lessive
22h - Bob dort
```

Après avoir lu les deux entiers, votre programme doit afficher ce que fait Bob à ce moment là. Par exemple, pour 17 34 on devrait voir Bob est au pub (happy hour!). Si le moment est à la fin d'un intervalle et au début d'un autre, affichez l'activité future. Par exemple pour 13 30 affichez Bob travaille.

Solution de l'exercice 4

```
1 #include <stdio.h>
2
3 int main() {
4     int heure, minute;
5
6     printf("Entrez l'heure et les minutes : ");
7     scanf("%d %d", &heure, &minute);
8
9     if (heure >= 0 && heure < 8 || heure == 8 && minute < 30) {
10        printf("Bob dort\n");
11    } else if (heure == 8 && minute >= 30) {
12        printf("Bob prend le petit déjeuner\n");
13    } else if (heure >= 9 && heure < 11 || heure == 11 &&
14        minute < 45) {
15        printf("Bob travaille\n");
16    } else if (heure == 11 && minute >= 45 || heure < 13) {
17        printf("Bob mange son repas de midi\n");
18    } else if (heure == 13 && minute < 30) {
19        printf("Bob boit son café\n");
20    } else if (heure == 13 && minute >= 30 || heure < 17) {
21        printf("Bob travaille\n");
22    }
```

```

21 } else if (heure >= 17 && heure < 18) {
22     printf("Bob est au pub (happy hour!)\n");
23 } else if (heure >= 18 && heure < 22) {
24     printf("Bob fait sa lessive\n");
25 } else {
26     printf("Bob dort\n");
27 }
28
29 return 0;
30 }

```

Exo5 Devinettes

Faites deviner à l'utilisateur un nombre aléatoire entre 1 et 10 en 3 essais. Après chaque essai vérifiez si la réponse est juste. Si ce n'est pas juste, indiquez par un message si l'essai est trop haut ou trop bas. Si après les trois essais l'utilisateur n'a pas réussi à deviner le nombre, affichez-le quand même.

Pour générer un nombre aléatoire entre 1 et 10 utilisez le code ci-dessous :

```

1 #include <stdlib.h>
2 #include <time.h>
3
4 int main()
5 {
6     srand(time(0)); // Seed the random number generator
7     int secret = rand() % 10 + 1; // aleatoire entre 1 et 10
8     ...
9 }

```

Solution de l'exercice 5

```

1 #include <stdio.h>
2 #include <stdlib.h>
3 #include <time.h>
4
5 int main() {
6     srand(time(0)); // Seed the random number generator
7     int secret = rand() % 10 + 1; // Random int between 1 and
8     10
9     int essai;
10    int attempts = 0;
11
12    printf("Devinez le nombre secret entre 1 et 10 !\n");

```

```

13 printf("Entrez votre estimation : ");
14 scanf("%d", &essai);
15 attempts++;
16
17 if (essai == secret) {
18     printf("Félicitations ! Vous avez deviné le nombre
19 secret en %d tentative(s) !\n", attempts);
20     return 0;
21 } else if (essai < secret) {
22     printf("Trop bas !\n");
23 } else {
24     printf("Trop haut !\n");
25 }
26
27 printf("Entrez votre estimation : ");
28 scanf("%d", &essai);
29 attempts++;
30
31 if (essai == secret) {
32     printf("Félicitations ! Vous avez deviné le nombre
33 secret en %d tentative(s) !\n", attempts);
34     return 0;
35 } else if (essai < secret) {
36     printf("Trop bas !\n");
37 } else {
38     printf("Trop haut !\n");
39 }
40
41 printf("Entrez votre estimation : ");
42 scanf("%d", &essai);
43 attempts++;
44
45 if (essai == secret) {
46     printf("Félicitations ! Vous avez deviné le nombre
47 secret en %d tentative(s) !\n", attempts);
48 } else {
49     printf("Désolé, vous n'avez pas deviné le nombre secret
50 . C'était %d.\n", secret);
51 }
52
53 return 0;
54 }

```