

Introduction aux algorithmes

Algorithmique et structures de données, 2022-2023

P. Albuquerque (B410), P. Künzli et O. Malaspinas (A401), ISC, HEPIA
2022-09-28

En partie inspirés des supports de cours de P. Albuquerque

Rappel (1/2)

- Quel est l'algorithme pour le calcul des nombres 1ers?

Rappel (1/2)

- Quel est l'algorithme pour le calcul des nombres premiers?

```
bool est_premier(int nombre) {  
    int i = 2; // bonne pratique!  
    while (i < nombre) { // penser à bien indenter!  
        if (0 == nombre % i) { // ça rend le code LISIBLE!  
            return false;  
        }  
        i += 1;  
    }  
    return true;  
}
```

Rappel (2/2)

- Quelles structures de contrôles avons nous vues?

Rappel (2/2)

- Quelles structures de contrôles avons nous vues?
- La boucle `while`,
- La boucle `do ... while`,
- La boucle `for`,
- La condition `if ... else if ... else`,

Exercice: la factorielle

Écrire un programme qui calcule la factorielle d'un nombre

$$N! = 1 \cdot 2 \cdot \dots \cdot (N - 1) \cdot N.$$

Par groupe de 3: écrire un pseudo-code

Exercice: la factorielle

Écrire un programme qui calcule la factorielle d'un nombre

$$N! = 1 \cdot 2 \cdot \dots \cdot (N - 1) \cdot N.$$

Par groupe de 3: écrire un pseudo-code

```
entier factorielle(entier n)
```

```
    i = 1
```

```
    fact = 1
```

```
    tant que i <= n
```

```
        fact *= i
```

```
        i += 1
```

```
    retourne fact
```

Exercice: la factorielle

Écrire un programme qui calcule la factorielle d'un nombre

$$N! = 1 \cdot 2 \cdot \dots \cdot (N - 1) \cdot N.$$

Par groupe de 3: écrire un code en C

Quand vous avez fini postez le code sur le salon matrix.

Exercice: la factorielle

Écrire un programme qui calcule la factorielle d'un nombre

$$N! = 1 \cdot 2 \cdot \dots \cdot (N - 1) \cdot N.$$

Par groupe de 3: écrire un code en C

Quand vous avez fini postez le code sur le salon matrix.

```
#include <stdio.h>
int main() {
    int nb = 10;
    int fact = 1;
    int iter = 2;
    while (iter <= nb) {
        fact *= iter;
        iter++;
    }
    printf("La factorielle de %d est %d\n", nb, fact);
}
```

Exercice: la factorielle

Écrire un programme qui calcule la factorielle d'un nombre

$$N! = 1 \cdot 2 \cdot \dots \cdot (N - 1) \cdot N.$$

Par groupe de 3: écrire un code en C

Quand vous avez fini postez le code sur le salon matrix.

```
#include <stdio.h>
int main() {
    int nb = 10;
    int fact = 1;
    int iter = 2;
    while (iter <= nb) {
        fact *= iter;
        iter++;
    }
    printf("La factorielle de %d est %d\n", nb, fact);
}
```

Comment améliorer ce code? (notez ça sur une feuille)

Exercice: la factorielle en mieux

Individuellement

1. Écrivez l'algorithme de calcul de deux façon différentes.
2. Que se passe-t-il si $n \geq 15$?
3. Pour celles et ceux qui ont fini pendant que les autres essaient: faites-le en récursif (sans aide).

Exercice: la factorielle en mieux

Individuellement

1. Écrivez l'algorithme de calcul de deux façon différentes.
2. Que se passe-t-il si $n \geq 15$?
3. Pour celles et ceux qui ont fini pendant que les autres essaient: faites-le en récursif (sans aide).

Postez vos solutions sur matrix!

Exercice: test si un nombre est premier

Avec tout ce que vous avez appris jusqu'ici:

- Écrivez le code testant si un nombre est premier.
- Quels sont les ajouts possibles par rapport au code de la semaine passée?
- Rencontrez-vous des problèmes éventuels de compilation?
- Voyez-vous une façon de générer des nombres premiers avec votre programme?

Exercice: test si un nombre est premier

Avec tout ce que vous avez appris jusqu'ici:

- Écrivez le code testant si un nombre est premier.
- Quels sont les ajouts possibles par rapport au code de la semaine passée?
- Rencontrez-vous des problèmes éventuels de compilation?
- Voyez-vous une façon de générer des nombres premiers avec votre programme?

Implémentez-la et postez votre code sur le salon matrix (10 min).

Corrigé: enfin pas vraiment mais juste un possible

```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <stdbool.h>
int main() {
    int nb = 1;
    printf("Entrez un nombre: ");
    scanf("%d", &nb);
    bool premier = true;
    for (int div = 2; div <= sqrt(nb); div++) {
        if (nb % div == 0) {
            premier = false;
            break;
        }
    }
    if (premier) {
        printf("Le nombre %d est premier\n", nb);
    } else {
        printf("Le nombre %d n'est pas premier\n", nb);
    }
    return 0;
}
```

Voyons quelques algorithmes supplémentaires

- Plus petit commun multiple (PPCM) de deux nombres
- Autre algorithme de calcul du PPCM de deux nombres
- Plus grand commun diviseur (PGCD) de deux nombres

Le calcul du PPCM (1/5)

Définition

Le plus petit commun multiple (PPCM), p , de deux entiers non nuls, a et b , est le plus petit entier strictement positif qui soit multiple de ces deux nombres.

Exemples:

$$\text{PPCM}(3, 4) = 12,$$

$$\text{PPCM}(4, 6) = 12,$$

$$\text{PPCM}(5, 15) = 15.$$

Le calcul du PPCM (1/5)

Définition

Le plus petit commun multiple (PPCM), p , de deux entiers non nuls, a et b , est le plus petit entier strictement positif qui soit multiple de ces deux nombres.

Exemples:

$$\text{PPCM}(3, 4) = 12,$$

$$\text{PPCM}(4, 6) = 12,$$

$$\text{PPCM}(5, 15) = 15.$$

Mathématiquement

Décomposition en nombres premiers:

$$36 = 2^2 \cdot 3^2, \quad 90 = 2 \cdot 5 \cdot 3^2,$$

On garde tous les premiers à la puissance la plus élevée

$$\text{PPCM}(36, 90) = 2^2 \cdot 3^2 \cdot 5 = 180.$$

Le calcul du PPCM (2/5)

Exemple d'algorithme

PPCM(36, 90):

36 < 90 // 36 + 36

72 < 90 // 72 + 36

108 > 90 // 90 + 90

108 < 180 // 108 + 36

144 < 180 // 144 + 36

180 = 180 // *The End!*

- 5 additions, 5 assignations, et 6 comparaisons.

Le calcul du PPCM (2/5)

Exemple d'algorithme

PPCM(36, 90) :

36 < 90 // 36 + 36

72 < 90 // 72 + 36

108 > 90 // 90 + 90

108 < 180 // 108 + 36

144 < 180 // 144 + 36

180 = 180 // *The End!*

- 5 additions, 5 assignations, et 6 comparaisons.

Transcrivez cet exemple en algorithme (groupe de 3), 5min

Le calcul du PPCM (2/5)

Exemple d'algorithme

PPCM(36, 90) :

36 < 90 // 36 + 36

72 < 90 // 72 + 36

108 > 90 // 90 + 90

108 < 180 // 108 + 36

144 < 180 // 144 + 36

180 = 180 // *The End!*

- 5 additions, 5 assignations, et 6 comparaisons.

Transcrivez cet exemple en algorithme (groupe de 3), 5min

et codez-le!

Le calcul du PPCM (3/5)

Tentative de correction

```
int main() {
    int m = 15, n = 12;
    int mult_m = m, mult_n = n;
    while (mult_m != mult_n) {
        if (mult_m > mult_n) {
            mult_n += n;
        } else {
            mult_m += m;
        }
    }
    printf("Le ppcm de %d et %d est %d\n", n, m, mult_m);
}
```

Le calcul du PPCM (3/5)

Tentative de correction

```
int main() {
    int m = 15, n = 12;
    int mult_m = m, mult_n = n;
    while (mult_m != mult_n) {
        if (mult_m > mult_n) {
            mult_n += n;
        } else {
            mult_m += m;
        }
    }
    printf("Le ppcm de %d et %d est %d\n", n, m, mult_m);
}
```

- Combien d'additions / comparaisons au pire?

Le calcul du PPCM (4/5)

Réusinage: Comment décrire une fonction qui ferait ce calcul (arguments, sorties)?

Le calcul du PPCM (4/5)

Réusinage: Comment décrire une fonction qui ferait ce calcul (arguments, sorties)?

En C on pourrait la décrire comme

```
int ppcm(int a, int b); // La **signature** de cette fonction
```

Le calcul du PPCM (4/5)

Réusinage: Comment décrire une fonction qui ferait ce calcul (arguments, sorties)?

En C on pourrait la décrire comme

```
int ppcm(int a, int b); // La **signature** de cette fonction
```

Algorithme

Par groupe de 3 (5-10min):

- réfléchissez à un algorithme alternatif donnant le PPCM de deux nombres;
- écrivez l'algorithme en pseudo-code.

Le calcul du PPCM (5/5)

Indication

Si un nombre, p , est multiple de a et de b alors il peut s'écrire $p = a * i = b * j$ ou encore $p / a = i$ et $p / b = j$.

Le calcul du PPCM (5/5)

Indication

Si un nombre, p , est multiple de a et de b alors il peut s'écrire $p = a * i = b * j$ ou encore $p / a = i$ et $p / b = j$.

Pseudo-code

```
int ppcm(int a, int b) {  
    for (i in [1, b]) {  
        if a * i est divisible par b {  
            return a * i  
        }  
    }  
}
```

Le code du PPCM de 2 nombres (1/2)

Implémentez le pseudo-code et postez le code sur matrix (5min).

Le code du PPCM de 2 nombres (1/2)

Implémentez le pseudo-code et postez le code sur matrix (5min).

Un corrigé possible

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int n = 15, m = 12;
    int i = 1;
    while (n * i % m != 0) {
        i++;
    }
    printf("Le ppcm de %d et %d est %d\n", n, m, n*i);
}
```

Le code du PPCM de 2 nombres (2/2)

Corrigé alternatif

```
#include <stdio.h>
#include <stdlib.h>
int main() {
    int res = n*m;
    for (int i = 2; i <= m; i++) {
        if (n * i % m == 0) {
            res = n * i;
            break;
        }
    }
    printf("Le ppcm de %d et %d est %d\n", n, m, res);
}
```