

- Algorithmme: Exercices Résolus -

Bekkali Hamza

January 30, 2025

1 Exercices boucle for

Exercice 1

Écrire un algorithme qui lit un entier N et affiche les entiers de 1 à N .

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i;
5     // Exercice 1
6     printf("Entrez N : ");
7     scanf("%d", &N);
8     for (i = 1; i <= N; i++) {
9         printf("%d\n", i);
10    }
11    return 0;
12 }
```

Exercice 2

Écrire un algorithme qui lit un entier N et affiche les nombres pairs de 2 à N .

Correction

```
1 #include <stdio.h>
2 int main() {
3     int N, i;
4     // Exercice 2
5     printf("Entrez N : ");
6     scanf("%d", &N);
7     for (i = 2; i <= N; i += 2) {
8         printf("%d\n", i);
9     }
10    return 0;
11 }
```

Exercice 3

Écrire un algorithme qui lit un entier N et affiche les entiers de 1 à N dans l'ordre décroissant.
Exemple : $N = 5$ Écran : 5, 4, 3, 2, 1.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i;
5     printf("Entrez N : ");
6     scanf("%d", &N);
7     for (i = N; i >= 1; i--) {
8         printf("%d\n", i);
9     }
10    return 0;
11 }
```

Exercice 4

Écrire un algorithme qui lit 3 entiers A , B et C , et affiche les diviseurs de C qui se trouvent dans l'intervalle $[A, B]$. Exemple : $A = 5$, $B = 23$, $C = 40$ Écran : 5, 8, 10, 20.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int A, B, C, i;
5     printf("Entrez A : ");
6     scanf("%d", &A);
7     printf("Entrez B : ");
8     scanf("%d", &B);
9     printf("Entrez C : ");
10    scanf("%d", &C);
11    for (i = A; i <= B; i++) {
12        if (C % i == 0) {
13            printf("%d\n", i);
14        }
15    }
16    return 0;
17 }
```

Exercice 5

Modifier l'algorithme de l'exercice 4 pour calculer et afficher le nombre des diviseurs de C . Exemple : $A = 5$, $B = 23$, $C = 40$ Écran : 4.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int A, B, C, i, compteur = 0;
5
6     printf("Entrez A : ");
7     scanf("%d", &A);
8     printf("Entrez B : ");
9     scanf("%d", &B);
10    printf("Entrez C : ");
11    scanf("%d", &C);
12    for (i = A; i <= B; i++) {
13        if (C % i == 0) {
14            compteur++;
15        }
16    }
17    printf("Nombre de diviseurs : %d\n", compteur);
18    return 0;
19 }
```

Exercice 6

Écrire un algorithme qui détermine si un entier N est premier ou non.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, premier = 1;
5
6     printf("Entrez N : ");
7     scanf("%d", &N);
8     if (N <= 1) {
9         printf("N n'est pas premier\n");
10    } else {
11        for (i = 2; i <= N / 2; i++) {
12            if (N % i == 0) {
13                premier = 0;
14                break;
15            }
16        }
17        if (premier) {
18            printf("N est premier\n");
19        } else {
20            printf("N n'est pas premier\n");
21        }
22    }
23    return 0;
24 }
```

Exercice 7

Écrire un algorithme qui lit un entier N ($N \geq 1$) et calcule la somme $S = 1 + 2 + 3 + \dots + N$.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, S = 0;
5     printf("Entrez N : ");
6     scanf("%d", &N);
7     for (i = 1; i <= N; i++) {
8         S += i;
9     }
10    printf("La somme est : %d\n", S);
11
12    return 0;
13 }
```

Exercice 8

Écrire un algorithme qui affiche chaque entier de $[100, 999]$ qui est égal à la somme des cubes de ses chiffres.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int i, somme, temp, chiffre;
5     for (i = 100; i <= 999; i++) {
6         somme = 0;
7         temp = i;
8         while (temp > 0) {
9             chiffre = temp % 10;
10            somme += chiffre * chiffre * chiffre;
11            temp /= 10;
12        }
13        if (somme == i) {
14            printf("%d\n", i);
15        }
16    }
17
18    return 0;
19 }
```

Exercice 9

Écrire un algorithme qui lit N entiers (un par un) et calcule leur somme S . Exemple : $N = 5$, valeurs saisies : 7, 3, -2, 4, 10 $S = 22$.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, entier, S = 0;
5     printf("Entrez N : ");
6     scanf("%d", &N);
7     for (i = 1; i <= N; i++) {
8         printf("Entrez un entier : ");
9         scanf("%d", &entier);
10        S += entier;
11    }
12    printf("La somme est : %d\n", S);
13
14    return 0;
15 }
```

Exercice 10

Écrire un algorithme qui lit N entiers et calcule le nombre des pairs et celui des impairs. Exemple : $N = 10$, valeurs saisies : 0, 7, 2, 1, 4, 8, 20, 11, 14, 30 $NP = 6$, $NI = 4$.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, entier, NP = 0, NI = 0;
5
6     printf("Entrez N : ");
7     scanf("%d", &N);
8     for (i = 1; i <= N; i++) {
9         printf("Entrez un entier : ");
10        scanf("%d", &entier);
11        if (entier % 2 == 0) {
12            NP++;
13        } else {
14            NI++;
15        }
16    }
17    printf("Nombre de pairs : %d\n", NP);
18    printf("Nombre d'impairs : %d\n", NI);
19
20    return 0;
21 }
```

Exercice 11

Écrire un algorithme qui lit N entiers et détermine le maximum. Exemple : $N = 6$, entiers saisis : 7, 3, 4, 11, 8, 1 Max = 11.

Correction

```
1 #include <stdio.h>
2 int main() {
3     int N, i, entier, Max;
4     printf("Entrez N : ");
5     scanf("%d", &N);
6     Max = 0;
7     for (i = 1; i <= N; i++) {
8         printf("Entrez un entier : ");
9         scanf("%d", &entier);
10        if (entier > Max) {
11            Max = entier;
12        }
13    }
14    printf("Le maximum est : %d\n", Max);
15    return 0;
16 }
```

Exercice 12

Modifier l'algorithme de l'exercice 11 pour déterminer le maximum et son nombre d'occurrences (répétitions).

Exemple : $N = 10$, entiers saisis : 5, 3, 5, 2, 7, 4, 7, 7, 1, 6 Max = 7, Nb = 3.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, i, entier, Max, Nb;
5
6     printf("Entrez N : ");
7     scanf("%d", &N);
8     Max = 0;
9     Nb = 0;
10    for (i = 1; i <= N; i++) {
11        printf("Entrez un entier : ");
12        scanf("%d", &entier);
13        if (entier > Max) {
14            Max = entier;
15            Nb = 1;
16        } else if (entier == Max) {
17            Nb++;
18        }
19    }
20    printf("Le maximum est : %d\n", Max);
21    printf("Le nombre d'occurrences est : %d\n", Nb);
22
23    return 0;
24 }
```

2 Exercices boucle While(...)

Exercice 1

Écrire un algorithme qui lit deux entiers positifs A et B ($B \neq 0$) et calcule le quotient de la division entière de A par B sans utiliser les opérateurs : Div, Mod, / et *. Exemple : A = 17, B = 5 Q = 3.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int A, B, Q, T;
5     printf("Entrez A : ");
6     scanf("%d", &A);
7     printf("Entrez B (B      0) : ");
8     scanf("%d", &B);
9     Q = 0;
10    T = A;
11    while (T >= B) {
12        T = T - B;
13        Q = Q + 1;
14    }
15    printf("Le quotient est : %d\n", Q);
16    return 0;
17 }
```

Exercice 2

Modifier l'algorithme de l'exercice 1 pour déterminer aussi le reste de la division. Exemple : A = 17, B = 5 Q = 3, R = 2.

Correction

```
1 #include <stdio.h>
2 int main() {
3     int A, B, Q, T, R;
4     printf("Entrez A : ");
5     scanf("%d", &A);
6     printf("Entrez B (B      0) : ");
7     scanf("%d", &B);
8     Q = 0;
9     T = A;
10    while (T >= B) {
11        T = T - B;
12        Q = Q + 1;
13    }
14    R = T;
15    printf("Le quotient est : %d\n", Q);
16    printf("Le reste est : %d\n", R);
17    return 0;
18 }
```

Exercice 3

Écrire un algorithme qui décompose un entier N en facteurs premiers en affichant chaque facteur et sa puissance.

Correction

```
1 #include <stdio.h>
2 int main() {
3     int N, i, puissance;
4     printf("Entrez N : ");
5     scanf("%d", &N);
6     i = 2;
7     while (N > 1) {
8         puissance = 0;
9         while (N % i == 0) {
10             N = N / i;
11             puissance++;
12         }
13         if (puissance > 0) {
14             printf("Facteur : %d, Puissance : %d\n", i, puissance);
15         }
16         i++;
17     }
18     return 0;
19 }
```

Exercice 4

Écrire un algorithme qui lit un entier N et affiche ses chiffres en commençant par le dernier. Exemple : N = 74304 4, 0, 3, 4, 7.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, chiffre;
5     printf("Entrez N : ");
6     scanf("%d", &N);
7     while (N > 0) {
8         chiffre = N % 10;
9         printf("%d\n", chiffre);
10        N = N / 10;
11    }
12    return 0;
13 }
```


Exercice 5

Modifier l'algorithme de l'exercice 4 pour calculer la somme des chiffres de N. Exemple : N = 74304
S = 18.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, chiffre, S = 0;
5     printf("Entrez N : ");
6     scanf("%d", &N);
7     while (N > 0) {
8         chiffre = N % 10;
9         S = S + chiffre;
10        N = N / 10;
11    }
12    printf("La somme des chiffres est : %d\n", S);
13
14    return 0;
15 }
```

Exercice 6

Modifier l'algorithme de l'exercice 4 pour calculer le nombre des chiffres pairs de N et celui des impairs.
Exemple : N = 73498 NP = 2, NI = 3.

Correction

```
1 #include <stdio.h>
2
3 int main() {
4     int N, chiffre, NP = 0, NI = 0;
5
6     printf("Entrez N : ");
7     scanf("%d", &N);
8     while (N > 0) {
9         chiffre = N % 10;
10        if (chiffre % 2 == 0) {
11            NP++;
12        } else {
13            NI++;
14        }
15        N = N / 10;
16    }
17    printf("Nombre de chiffres pairs : %d\n", NP);
18    printf("Nombre de chiffres impairs : %d\n", NI);
19
20    return 0;
21 }
```

Exercice 7

Écrire un algorithme qui lit deux entiers X et Y , et calcule X^Y en utilisant une boucle et l'opération de multiplication.

Rappel : $P = X^Y = X \times X \times \dots \times X$ (Y fois), si $Y > 0$.

Correction

```
1 #include <stdio.h>
2 int main() {
3     int X, Y, P, i;
4     printf("Entrez X : ");
5     scanf("%d", &X);
6     printf("Entrez Y : ");
7     scanf("%d", &Y);
8     P = 1;
9     i = 0;
10    while (i < Y) {
11        P = P * X;
12        i++;
13    }
14    printf("X^Y est : %d\n", P);
15    return 0;
16 }
```

Exercice 8

Écrire un algorithme qui détermine les deux premiers entiers de 3 chiffres, à partir de 100, qui sont égaux à la somme de leurs chiffres au cube. Exemple : $153 = 1^3 + 5^3 + 3^3$.

Correction

```
1 #include <stdio.h>
2 int main() {
3     int N = 100, compteur = 0, somme, temp, chiffre;
4     while (compteur < 2) {
5         somme = 0;
6         temp = N;
7         while (temp > 0) {
8             chiffre = temp % 10;
9             somme = somme + (chiffre * chiffre * chiffre);
10            temp = temp / 10;
11        }
12        if (somme == N) {
13            printf("%d\n", N);
14            compteur++;
15        }
16        N++;
17    }
18
19    return 0;
20 }
```

Exercice 9

Écrire un algorithme qui lit N entiers et calcule le nombre de séquences croissantes parmi ces N entiers. Exemple : N = 9, les entiers saisis sont : 7, 8, 10, 2, 4, 1, 8, 14, 20. Résultat : NS = 3 (les séquences croissantes sont 7, 8, 10 ; 2, 4 ; 1, 8, 14, 20).

Correction

```
1  #include <stdio.h>
2
3  int main() {
4      int N, i, entier, precedent, NS = 0;
5      int croissant = 0;
6
7      printf("Entrez N : ");
8      scanf("%d", &N);
9      if (N > 0) {
10         printf("Entrez un entier : ");
11         scanf("%d", &entier);
12         precedent = entier;
13         N--;
14     }
15     while (N > 0) {
16         printf("Entrez un entier : ");
17         scanf("%d", &entier);
18         if (entier > precedent) {
19             if (croissant == 0) {
20                 NS++;
21                 croissant = 1;
22             }
23         } else {
24             croissant = 0;
25         }
26         precedent = entier;
27         N--;
28     }
29     printf("Nombre de s quences croissantes : %d\n", NS);
30
31     return 0;
32 }
```