

TD1 – Programmation en C

Exercice 1

Soit le programme C suivant :

```
#include <stdio.h>
int main()
{
    int N=10, P=5, Q=10, R;
    char C='S';

    N = 5; P = 2;
    Q = N++ > P || P++ != 3;
    1 printf ("C : N=%d P=%d Q=%d\n", N, P, Q);

    N = 5; P = 2;
    Q = N++ < P || P++ != 3;
    2 printf ("D : N=%d P=%d Q=%d\n", N, P, Q);

    N = 5; P = 2;
    Q = ++N == 3 && ++P == 3;
    3 printf ("E : N=%d P=%d Q=%d\n", N, P, Q);

    N=5; P=2;
    Q = ++N == 6 && ++P == 3;
    4 printf ("F : N=%d P=%d Q=%d\n", N, P, Q);

    N=C;
    5 printf ("G : %c %c\n", C, N);
    6 printf ("H : %d %d\n", C, N);
    7 printf ("I : %x %x\n", C, N);

    return 0;
}
```

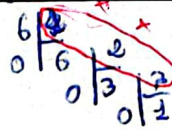
Qu'affiche le programme ci-dessus ?

Exercice 2

Écrire un programme C qui permette de saisir N notes et d'indiquer la moyenne de ces notes (N n'est pas connu).

Remarques :

- La saisie des notes se poursuit donc tant qu'une note incorrecte n'est pas détectée.
- Une note incorrecte est une note inférieure à 0 ou supérieure à 20.

**Exercice 3**

Écrire un programme qui affiche tous les diviseurs d'un nombre positif lit à partir du clavier
Modifiez le programme pour déterminer si ce nombre est parfait.

NB. Un nombre est parfait si la somme de ses diviseurs est égale à ce nombre

Exemple $6 = 1 + 2 + 3$

Exercice 4

Une phrase terminée par un point est saisie caractère par caractère à l'entrée standard.

Écrire un programme donnant le nombre d'occurrence des caractères "LE" dans cette phrase. Les traitements seront effectués "au vol", c'est-à-dire sans effectuer de stockage dans un tableau.

TD1 - Prog.C

EXERCICE 1 :

① 5 2 1

True (can 6 > 2)

$$\textcircled{1} = N + + > P \parallel P_{++}! = 3$$

② 631

③ 6 2 0

L → (car on fire) → $\lambda x^3 \text{ true} \rightarrow +P$

④	6	3	1
---	---	---	---

⑤ SS

⑥ 83 83

⑦ 53 53

Exercice 2:

①. E, N, note, i, S
bonnie Ventres

- `line(N) N > 0 do { scanf("%d", &N) } while`
- `line(note) do (note >= 0 el <= 20) while) for`

$$\begin{matrix} \uparrow \\ \{ s+ = note; \} \text{ for} \end{matrix}$$

D.B: $M = S/N$

```
Rep #include <stdio.h>
```

```
int main() {
```

```
float M, node, S;
```

Sent $i, N;$

```
do { scanf("%d", &N); } while (!N > 0);
```

```
for (i=0; i < N; i++) {
```

$$N \leq 0)$$


```

do { scanf ("%f", &note); } while ( !(note >= 0 && note <= 20) );
    note + note + note
    S = S + note;  => S += note
}
M = S / N;
printf ("la moyenne est : %.f", M);
return (0);
}

```

Exercice 3:

```

#include <stdio.h>
int main () {
    int N, Nb, i;
    do { printf ("donner un entier N : ");
        scanf ("%d", &N);
    } while ( ! N > 0 );
    Nb = 0;
    for (i = 1; i = N / 2; i++) {
        if (N % i == 0)
            printf ("la division de N par %d", i);
            Nb = Nb + i;
        if (Nb == N)
            printf ("le nombre est parfait");
        else
            return 0;
    }
    return 0;
}

```


TD2 – Programmation en C**Exercice 1**

Écrire un programme qui lit la taille N d'un tableau T du type entier, remplit le tableau par des valeurs entrées au clavier et affiche le tableau.

Créer ensuite un tableau $T1$ contenant les entiers pairs de T et un tableau $T2$ contenant les entiers impairs de T , et afficher les deux tableaux $T1$ et $T2$.

Exercice 2

Écrire un programme C qui demande à l'utilisateur de saisir les notes d'une classe et qui calcule et affiche ensuite le pourcentage de notes supérieures à la moyenne de la classe.

Exercice 3 : Programmation descendante

On donne une phrase P et un mot m . écrire un programme C qui permette de trouver si les lettres de m se trouvent dans le même ordre dans p .

Exercice 4 : Plateau

On possède une table de nombres entiers, classés par valeur croissante, chaque entier pouvant être répété plusieurs fois :

3 5 5 5 7 9 9 9 9 10 10 12 12 12 13

On appelle plateau la suite de tous les entiers consécutifs de même valeur : dans l'exemple ci-dessous, le plateau de valeur 5 comporte 3 nombres, celui de valeur 9 en comporte 4, celui de valeur 10 en comporte 2 etc. On appelle longueur d'un plateau le nombre de ses termes, et valeur d'un plateau la valeur commune de tous ses termes. Un vecteur a de n termes étant donné, trouver la longueur et la valeur de son plus long plateau.

TD2-Prog.C

Exercise 1:

```
#include <stdio.h>
int main() {
    int i, j = 0, k = 0, N;
    printf("N = ");
    scanf("%d", &N);
    while (N <= 0) {
        scanf("%d", &N);
    }

    int T[N], T1[N], T2[N];
    for (i = 0; i < N; i++) {
        scanf("%d", &T[i]);
        if (T[i] % 2 == 0) {
```

```
            T1[j] = T[i]; j++;
        } else {
            T2[k] = T[i]; k++;
        }
    }

    for (i = 0; i < N; i++) {
        printf("%d", T[i]);
    }
    for (i = 0; i < j; i++) {
        printf("%d", T1[i]);
    }
    for (i = 0; i < k; i++) {
        printf("%d", T2[i]);
    }

    return 0; }
```

Exercise 2:

```
#include <stdio.h>
int main() {
    float T[N];
    int N = 30;
    int nb = 0;
    int i, j;
    for (i = 0; i < N; i++) {
        scanf("%f", &T[i]);
    }
    while (! (T[i] >= 0 && T[i] <= 20));
```

```
    tot = T[i];
    Moy = tot / N;
    for (j = 0; j < N; j++) {
        if (T[j] > Moy) {
            nb++;
        }
    }
    printf("%f", nb / N * 100);
    return 0; }
```


Exercice 3:

```
#include <stdio.h>
```

```
int main () {
```

```
    int np, nm, i=0, j=0, n=0;
```

```
    np = strlen(p);
```

```
    nm = strlen(m);
```

```
    do { if (m[i] == p[j]) {
```

```
        i++;
```

```
        j++;
```

```
        n++;
```

```
    }
```

```
    else
```

```
        j++; } while (j != np || i != nm);
```

```
    if (n == m)
```

```
        printf("Les lettres de %s se trouvent dans le me ordre dans %s", m, p);
```

```
    else
```

```
        printf("Les lettres de %s ne se trouvent pas dans le me ordre  
dans %s, n p);
```

```
    return 0;
```

```
}
```