

# Solutions des exercices de cours : Manipulation des sous-programmes

---

1. Écrire une fonction qui augmente son argument par 1.

```
void increment(int &x){
    x++;
}

int main(){
    int n = 2;
    cout << "n = " << n << endl;
    increment(n);
    cout << "n = " << n << endl;
}
```

```
void increment(int *x){
    (*x)++;
}

int main(){
    int n = 5;
    cout << "n = " << n << endl;
    increment(&n);
    cout << "n = " << n << endl;
}
```

2. Écrire une fonction qui permet d'échanger les valeurs de deux variables entières.

```
void swap(int &x, int &y){
    int tmp = x;
    x = y;
    y = tmp;
}

int main(){
    int x = 5, y = 12;
    cout << "x = " << x << ", y = " << y << endl;
    swap(x, y);
    cout << "x = " << x << ", y = " << y << endl;
}
```

3. Écrire une fonction qui calcule la somme des nombres entiers de 1 à N.  
 $S = 1 + 2 + 3 + \dots + N$

```
int somme(int n){
    return (n * (n + 1) / 2);
}
```

4. Écrire une fonction appelée largestDigit(), qui retourne le plus grand chiffre de son argument entier.  
Par exemple : largestDigit(3185) retourne 8 et largestDigit(-65665) retourne 6.

```
#include <cmath>

int largestDigit(int n){
    int largest = 0;
    n = abs(n);
    while(n > 0){
        int digit = n % 10;
        if(largest < digit)
            largest = digit;
        n /= 10;
    }
    return largest;
}
```

```
int main(){
    int number;
    cout << "Enter an integer: ";
    cin >> number;
    cout << "The largest digit of " << number << " is: "
        << largestDigit(number) << endl;
}
```

4. Écrire une fonction qui prend un tableau d'entiers comme argument et retourne le plus grand nombre dans ce tableau.

```
const int N = 10;

int largestInt(int a[]){
    int max = a[0];
    for(int i = 1; i < N; i++){
        if(max < a[i]) { max = a[i]; }
    }

    return max;
}

int main(){
    int array[N] = {10, 1, 12, -56, 1000, 110, 21, 120, 56, 30};
    cout << "Max = " << largestInt(array) << endl;
}
```

```
const int N = 10;
typedef int array[N];

int largestInt(array a){
    int max = a[0];
    for(int i = 1; i < N; i++){
        if(max < a[i]){ max = a[i]; }
    }

    return max;
}

int main(){
    array t = {10, 1, 12, -56, 900, 110, 21, 120, 56, 30};
    cout << "Max = " << largestInt(t) << endl;
}
```

5. Écrire une fonction qui prend un tableau d'entiers non négatifs comme argument et retourne le deuxième plus grand nombre entier dans ce tableau. La fonction retourne -1 s'il n'y a pas de deuxième plus grand nombre dans le tableau.

Si le tableau est	La fonction retourne
{1, 2, 3, 4}	3
{4, 1, 2, 3}	3
{1, 1, 2, 2}	1
{1, 1}	-1
{1}	-1
{}	-1

```

const int N = 10;

int secondLargestInt(unsigned a[]){
    int max1 = a[0];
    int max2 = -1;
    for(int i = 1; i < N; i++){
        if(max1 < a[i]){
            max2 = max1;
            max1 = a[i];
        }
        else if (max1 > a[i] && max2 < a[i])
            max2 = a[i];
    }
    return max2;
}

int main(){
    unsigned array[N] = {10, 1, 12, 56, 100, 100, 21, 120, 56, 30};
    cout << "Second max = " << secondLargestInt(array) << endl;
}

```

**Remarque :** La fonction doit retourner un entier, donc on peut pas écrire:

`unsigned secondLargestInt(unsigned a[]);`

La signature de la fonction est : `int secondLargestInt(unsigned a[]);`