

TP_s6.2: struct (solution)

Lien avec le [MOOC Initiation à la Programmation \(en C++\)](#)

Exercices semaine6.2 du MOOC : 2^{de} partie struct

La solution des exercices du MOOC est disponible sur le MOOC.

Exercice Complémentaire (ExC)

ExC 4 : Using C++ min() and max() functions (in English)

ExC4 Exercise : read 4 values (either integer, floating point or character) and use the functions to get the min and the max of those 4 values.

```
#include <iostream>
#include <algorithm>
using namespace std;

int main ()
{
    int a,b,c,d;
    cin >> a >> b >> c >> d ;

    cout << endl << "leur minimum est: " << min(min(a,b),min(c,d)) << endl;
    cout << endl << "leur maximum est: " << max(max(a,b),max(c,d)) << endl;

    return 0;
}
```

ExC5 : opérateurs bit à bit

fonctions de manipulation à l'échelle du bit ou groupe de bits.

```
unsigned int f1(unsigned int n, unsigned int index)
{
    if(index < 32)
    {
        return (n >> index) & 1;
    }

    cout << "Index must be within [0, 31] instead of : "
         << index << endl;

    exit(0);
}
```

```

unsigned int f2(unsigned int n, unsigned int index,
               unsigned int val)
{
    if(index < 32 and val < 2)
    {
        return (n & ~(1<<index)) | (val << index);
    }

    cout << "Index must be within [0, 31] "
          " and val must be only 0 or 1" << endl;

    exit(0);
}

```

```

unsigned int f3(unsigned int n, unsigned int width,
               unsigned int shift, unsigned int val)
{
    if( 1< width and width < 32 and
        shift < 32-width and val < (1<<width))
    {
        unsigned int mask((1<<width) -1);
        return (n & ~(mask<<shift)) | (val << shift);
    }

    cout << "width must be within [1, 32], "
          " shift within [0, 32-width]"
          " and val within [0, 2^width-1]" << endl;

    exit(0);
}

```