

Devoir : le Challenge des paramètres

```
#include <iostream>

using namespace std;

void testerFunction(int* params, size_t size) {
    cout << "Function called with " << size << " parameters." << endl;
}

int main() {
    size_t i = 1;

    while (true) {
        int* params = new int[i];
        if (!params) {
            cout << "Memory allocation error." << endl;
            break;
        }

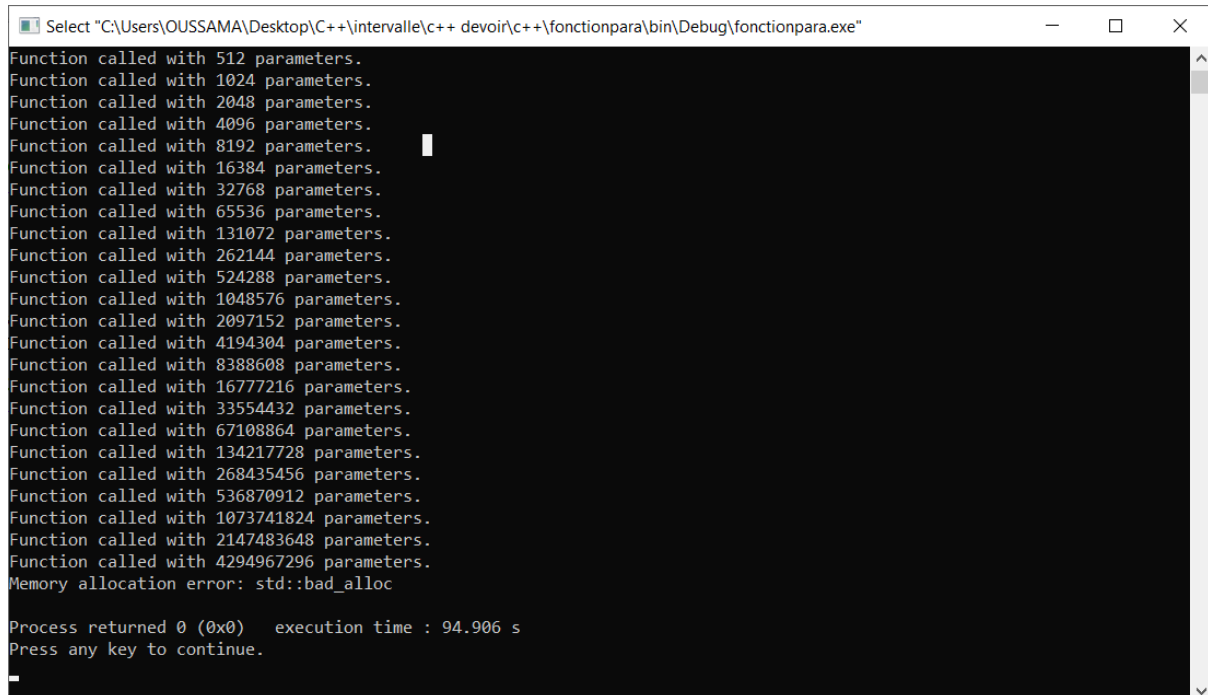
        testerFunction(params, i);

        delete[] params;

        i *= 2;
    }

    return 0;
}
```

Après exécuter ce code :



```
Select "C:\Users\OUSSAMA\Desktop\C++\intervalle\c++ devoir\c++\fonctionpara\bin\Debug\fonctionpara.exe"
Function called with 512 parameters.
Function called with 1024 parameters.
Function called with 2048 parameters.
Function called with 4096 parameters.
Function called with 8192 parameters.
Function called with 16384 parameters.
Function called with 32768 parameters.
Function called with 65536 parameters.
Function called with 131072 parameters.
Function called with 262144 parameters.
Function called with 524288 parameters.
Function called with 1048576 parameters.
Function called with 2097152 parameters.
Function called with 4194304 parameters.
Function called with 8388608 parameters.
Function called with 16777216 parameters.
Function called with 33554432 parameters.
Function called with 67108864 parameters.
Function called with 134217728 parameters.
Function called with 268435456 parameters.
Function called with 536870912 parameters.
Function called with 1073741824 parameters.
Function called with 2147483648 parameters.
Function called with 4294967296 parameters.
Memory allocation error: std::bad_alloc

Process returned 0 (0x0) execution time : 94.906 s
Press any key to continue.
■
```

conclusion :

Alors, on remarque qu'il y a un message d'erreur : **“Memory allocation error.”** Après 4 294 967 296 paramètres, la fonction peut accepter 4 294 967 296 paramètres. Cependant, pour une meilleure expérience, nous ne devrions pas utiliser un grand nombre de paramètres.