

Les tests

Vérifier automatiquement que le programme est correct : pas de bug, pas d'erreur de calcul, répond au besoin

plusieurs types de tests. Test Driven Development (TDD). Non régression

Les tests unitaires

- boost.test
- QTest
- CTest
- Autres

main.cpp

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

double eval(string const&) {
    return 1.0;
}

int main() {
    using Test = pair<string, double>;
    using Tests = Test[];
    const Tests tests {
        { "1 1 +", 2.0 },
        { "1 1 -", 0.0 },
        { "1 1 *", 1.0 },
        { "1 1 /", 1.0 }
    };
    for (auto test: tests) {
        const auto result = eval(test.first);
```

```

        cout << "\"" << test.first << "\" == " << result <<
        (test.second == result ? " [pass]" : " [fail]") << endl;
    }

    return 0;
}

```

affiche :

main.cpp

```

"1 1 +" == 1 [fail]
"1 1 -" == 1 [fail]
"1 1 *" == 1 [pass]
"1 1 /" == 1 [pass]

```

Idée d'exos : proposer un code avec erreurs pas forcément visibles.
Ecrire des tests permettant d'identifier les erreurs.

Les tests fonctionnels

Comment établir un scénario de test ? Comment générer des données ?

Chapitre précédent	Sommaire principal	Chapitre suivant
---------------------------	---	-------------------------

[Cours, C++](#)