



```
#include <iostream>
#include <numeric>
#include <cmath>

int main()
{
    double a = std::numeric_limits<double>::infinity();

    if (std::isinf(a))
        std::cout << "Inf detected.\n";
    else
        std::cout << "Inf NOT detected.\n";

    if (std::isfinite(a))
        std::cout << "Is finite.\n";
    else
        std::cout << "Is NOT finite.\n";

    return 0;
}
```

<https://godbolt.org/z/Y56o6ffE4>

-ffast-math

Inf NOT detected.

Is finite.

Clang trunk: `-std=c++2b -03`

`Inf detected.`

`Is NOT finite.`

```

#include <iostream>
#include <numeric>
#include <cmath>

int main()
{
    double a = std::numeric_limits<double>::infinity();

    if (std::isinf(a))
        std::cout << "Inf detected.\n";
    else
        std::cout << "Inf NOT detected.\n";

    if (std::isfinite(a))
        std::cout << "Is finite.\n";
    else
        std::cout << "Is NOT finite.\n";

    return 0;
}

```

Clang trunk: -std=c++2b -03

```

Inf detected.
Is NOT finite.

```

-ffast-math

```

Inf NOT detected.
Is finite.

```

```
// Return 256 random floats +-1.0f
static const std::vector<float>& getBufferData();

static void TransformXXX(benchmark::State& state)
{
    const auto& v = getBufferData();
    auto copy = v;

    for (auto _ : state)
    {
        std::transform (v.begin(), v.end(), copy.begin(), copy.end(),
                        [] (auto v1, auto v2) { return v1 + v2; });
    }
}
BENCHMARK(TransformXXX);
```