C++ WAT

Piotr Padlewski

C++ WAT

assert(map["Hello world!"] == 'e');

int map = 1;
int t[100];
42[t]; // the same as t[42]

C++ WAT

```
int main(){<:]()<%[](){[:>()<%}();}();}();}
```

```
int main(){<:]()<%
        [](){
          [:>()<%}();
        }();
      }();
}
```

```
int main(){[](){
        [](){
          [](){}();
        }();
      }();
}
```

C++ WAT

```cpp
void foo() {
    http://cpp.mimuw.edu.pl/
    printf("WAT??!");
    int n = 5;
    while(n --> 0) {
        //stuff
    }
    return (void)"Everything is fine";
}
int main() {
    foo();
}
```

WAT|

```cpp
struct A : public B
{
    A() try : B()
    {
        // constructor body
    }
    catch (...)
    {
        // exceptions from the initializer list are caught here
        // but also rethrown after this block (unless the program is aborted)
    }
};
```

```cpp
int main(){
    if (42 == 42) try {
        A a;
    }
    catch(...)
    {
    }
}
```

```cpp
typedef long long ll;

void foo(unsigned ll) {
    std::cout << "1\n";
}
void foo(unsigned long long) {
    std::cout << "2\n";
}
int main() {
    foo(2ull);
}
```

Since signed, unsigned, long, and short by default imply int, a type-name appearing after one of those specifiers is treated as the name being (re)declared."

```cpp
std::vector<std::string> v{{"testing", "123"}};
```

destruktor inta

```cpp
int main() {
    using int_ = int;

    int_ i;
    i.~int_();
}
```


WAT

```cpp
std::min(2000000000, 2100000000);
std::min(2000000000, 3000000000);
std::min(2200000000, 3000000000);

error: no matching function for call to 'min(int, long int)'
 std::min(2000000000, 3000000000);



   int func(int x);
   func((1, 2, 3, 4, 5));
```

```cpp
#include <iostream>
struct X {
  X() { std::cout << "X"; }
};

struct Y {
  Y(const X &x) { std::cout << "Y"; }
  void f() { std::cout << "f"; }
};

int main() {
  Y y(X());
  y.f();
}
```

weird.cc:15:11: error: request for member 'f' in 'y', which is of non-class type 'Y(X (*)())'
      y.f();

switch for

```cpp
int main() {
    int a = 2;
    int i;
    for (i = 0 ; i < 10 ; i++) {
        switch (a) {
            case 1:
                std::cout << "foo" << std::endl;
                break;
            case 2:
                std::cout << "foo2" << std::endl;
                break;
            default:
                std::cout << "bar" << std::endl;
        }
    }
}
```

```
foo2
foo2
foo2
foo2
foo2
foo2
foo2
foo2
foo2
foo2
```

switch for

```cpp
int main() {
    int a = 2;
    int i;
    switch (a) {
        case 1:
            for (i = 0 ; i < 10 ; i++)
                std::cout << "foo" << std::endl;
            break;
        case 2:
            for (i = 0 ; i < 10 ; i++)
                std::cout << "foo2" << std::endl;
            break;
        default:
            for (i = 0 ; i < 10 ; i++)
                std::cout << "bar" << std::endl;
    }
}
```

```
foo2
foo2
foo2
foo2
foo2
foo2
foo2
foo2
foo2
foo2
```

switch for

```cpp
int main() {
    int a = 2;
    int i;
    switch (a) {
        for (i = 0 ; i < 10 ; i++) {
            case 1:
                std::cout << "foo" << std::endl;
                continue;
            case 2:
                std::cout << "foo2" << std::endl;
                continue;
            default:
                std::cout << "bar" << std::endl;
        }
    }
}
```

clang out:
foo2

gcc out:
foo2
foo
foo
foo
foo
foo
foo
foo
foo
foo

optymalizacje UB

```
int table[4];
bool exists_in_table(int v)
{
    for (int i = 0; i <= 4; i++) {
        if (table[i] == v) return true;
    }
    return false;
}
```

```
int table[4];
bool exists_in_table(int v)
{
    return true;
}
```

fermat

```c
int fermat (void)
{
  const int MAX = 1000;
  int a=1,b=1,c=1;
  while (1) {
    if (((a*a*a) == ((b*b*b)+(c*c*c)))) return 1;
    a++;
    if (a>MAX) {
      a=1;
      b++;
    }
    if (b>MAX) {
      b=1;
      c++;
    }
    if (c>MAX) {
      c=1;
    }
  }
  return 0;
}
```

fermat

```c
#include <stdio.h>
int main (void)
{
  if (fermat()) {
    printf ("Fermat's Last Theorem has been disproved.\n");
  } else {
    printf ("Fermat's Last Theorem has not been disproved.\n");
  }
  return 0;
}
```

Fermat's Last Theorem has been disproved.

Weird error

```cpp
class StreamProcessor : public ga::ProcessorWriter
{
protected:
    bool doWrite_(Event &event);
    void doCommit_();
    std::unordered_set<int64_t> goodVisitors_, badVisitors_;
};
```

Weird error

```cpp
void StreamProcessor::doCommit_() {
  std::vector <int64_t> goodVisitors(goodVisitors.begin(),
                                     goodVisitors.end()),
      badVisitors(badVisitors.begin(), badVisitors.end());

  std::sort(goodVisitors.begin(), goodVisitors.end());
  std::sort(badVisitors.begin(), badVisitors.end());

  std::vector<int64_t> inter(badVisitors.size());
  auto it = std::set_difference(badVisitors.begin(), badVisitors.end(),
                                goodVisitors.begin(), goodVisitors.end(),inter.begin());
  inter.resize(it - inter.begin());
  for (auto val : inter)
    std::cout << val << std::endl;
}
```

Weird error

1. Program received signal SIGSEGV, Segmentation fault.
2. __memmove_ssse3_back () at ../sysdeps/x86_64/multiarch/memcpy-ssse3-back.S:2572
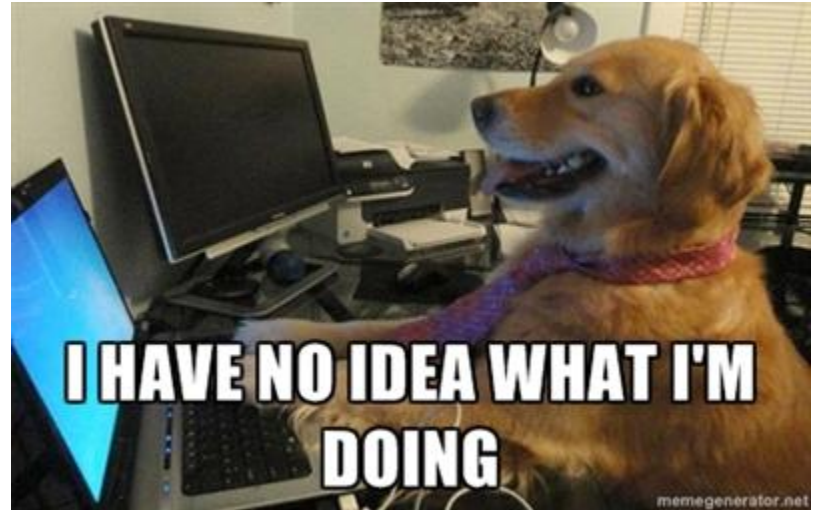3. 2572   ../sysdeps/x86_64/multiarch/memcpy-ssse3-back.S: Nie ma takiego pliku ani katalogu.

```cpp
#include <cstdio>
int main() {

    long long a = (long long)&a;

    scanf("%lld", a);

    printf("%lld\n", a); //works fine lol
}
```


I HAVE NO IDEA WHAT I'M DOING
memegenerator.net

Dzięki

http://kukuruku.co/hub/cpp/undefined-behavior-and-fermats-last-theorem

http://cppquiz.org/

http://www.reddit.com/r/cpp/comments/2ycbmj/c_wtfs/