```
#include <iostream>

void println(size_t v) {
    std::cout << v << ’ ’;
}

void println(bool b) {
    std::cout << std::boolalpha << b << ’ ’;
}

int main() {
    println(sizeof(bool));
    println(sizeof(char));
    println(sizeof(short));
    println(sizeof(int));
    println(sizeof(long));
    println(sizeof(long long));
    println(sizeof(float));
    println(sizeof(double));
    println(sizeof(long double));

    println(sizeof(char) == 1);
    println(sizeof(bool) < sizeof(int));
    println(sizeof(short) < sizeof(int));
    println(sizeof(int) == sizeof(long));
    println(sizeof(unsigned int) == sizeof(signed int));
    println(sizeof(long) >= 4);
}
```

What will this code print out?

```cpp
#include <iostream>

void p1(int i) { std::cout << i << std::endl; }

template<typename T> void p2(T x) { std::cout << x << std::endl; }

int main() {

    char c = 128;
    p1(c);
    p2(c);

    unsigned int i = -1;
    p1(i);
    p2(i);

    long l = -1;
    p1(l);
    p2(l);

    double f = 3.14;
    p1(f);
    p2(f);

    bool b = -1;
    p1(b);
    p2(b);
}
```

What might happen if you compile, link and run this program?

```cpp
#include <iostream>
#include <string>

enum day { mon, tue, wed, thu, fri, sat, sun };

std::string toString(day d) {
    std::string s;
    switch(d) {
    case mon: return "mon";
    case tue: return "tue";
    case wed: return "wed";
    case thu: return "thu";
    case sat: return "sat";
    case sun: return "sun";
    }
    return "x";
}

int main() {
    std::cout << toString(mon) << std::endl;
    std::cout << toString(wed) << std::endl;
    std::cout << toString(day(4)) << std::endl;
    std::cout << toString(day(8)) << std::endl;
}
```

What might happen if you compile, link and run this program?

```cpp
#include <iostream>

int main() {
    char* str = "Hello World\n";
    char* iter, end;

    iter = &str[0];
    end = &str[strlen(str)];

    while ( iter != end ) {
        std::cout.put(*iter);
        ++iter;
    }
    std::cout.flush();
}
```

What might happen if you try to compile, link and run this code?

```cpp
#include <iostream>

int x = 42;

int main() {
    int x = 43;
    {
        int x = 44;
        std::cout << x << std::endl;
    }
}
```

What might happen if you try to compile, link and run this code?

```cpp
#include <iostream>

int a;

namespace {
    int b;
}

int main() {
    static int c;
    int d;
    int * e = new int();

    std::cout << a << std::endl;
    std::cout << b << std::endl;
    std::cout << c << std::endl;
    std::cout << d << std::endl;
    std::cout << *e << std::endl;
}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

void p(int * p) {
    std::string separator = "";
    for ( int i = 0 ; i < 4 ; ++i ) {
        std::cout << separator << p[i];
        separator = ",";
    }
    std::cout << std::endl;
}

int main() {
    int a[4];
    int b[4] = {1,2};
    int c[4] = {};
    static int d[4];
    static int e[] = {1,2,3,4};

    p(a);
    p(b);
    p(c);
    p(d);
    p(e);
}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

int main() {
    char a[] = "Foo";
    char * b = "Bar";

    std::cout << a << " " << b << std::endl;

    a[0] = ’Z’;
    std::cout << a << " " << b << std::endl;

    b[0] = ’C’;
    std::cout << a << " " << b << std::endl;
}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

int main() {
    char a[] = "Hello " "World";

    for ( int i=0; a[i] != 0; ++i ) {
        std::cout.put(a[i]);
    }
    std::cout.put(’\n’);

    for ( char * p = a; *p != 0; ++p ) {
        std::cout.put(*p);
    }
    std::cout << std::endl;
}
```

What might happen if you try to compile, link and run this code?

```
int main() {
    int i = 42;

    int * const a = &i;
    *a = 43;
    a = &i;

    const int * b = &i;
    *b = 44;
    b = a;

    const int * const c = &i;
    *c = 45;
    c = a;
}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

struct X { char * a; char b[6]; int c; };

std::ostream & operator<<(std::ostream & os, const X & x) {
    return os << x.a << " " << x.b << " " << x.c;
}

int main() {
    X x = {"Hello", "World", 42};
    std::cout << x << std::endl;
    std::cout << sizeof(X) << std::endl;
}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

int main(int argc, const char ** argv) {
    std::cout << argc;
    for ( const char ** p = argv; *p != NULL; ++p ) {
        std::cout << " " << (*p);
    }
    std::cout << std::endl;
}
```

Given that this code is compiled, linked and executed like this:

```
  g++ -o foo foo.cpp
  ./foo bar gaz
```

What will be printed out. Please comment the code.

```cpp
#include <iostream>

int foo(int a) {
    std::cout << a;
    return a;
}

void bar(int b, int c) {
    std::cout << b << c;
}

int main() {
    int x = foo(5) + foo(3);
    foo(x);

    int y[4] = {};
    int i=1;
    y[i] = i++;
    foo(y[1]);

    bar(i++, i++);
}
```

What might happen if you try to compile, link and run this code?

```cpp
#include <iostream>

int main() {

    int x = 4;
    if ( 2 <= x <= 8 ) {
        std::cout << "a" << std::endl;
    } else {
        std::cout << "b" << std::endl;
    }

    if( x == 12 & 7 ) {
        std::cout << "c" << std::endl;
    } else {
        std::cout << "d" << std::endl;
    }

    if( x = 4 ) {
        std::cout << "e" << std::endl;
    } else {
        std::cout << "f" << std::endl;
    }

}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

class Foo {
    int value;
public:
    Foo() : value(42) { std::cout << "a"; }
    ~Foo() { std::cout << "b"; }
    Foo(const Foo & f) { std::cout << "c"; value = f.value; }
    Foo & operator=(const Foo & f) { std::cout << "d"; value = f.value; return *this;
    Foo operator++(int) { std::cout << "e"; Foo old(*this); ++*this; return old;}
    Foo & operator++() { std::cout << "f"; value += 4; return *this; }
};

int main() {
    Foo f1;
    std::cout << "-";
    ++f1;
    std::cout << "-";
    f1++;
    std::cout << "-";
    Foo f2 = f1;
    std::cout << "-";
    f2 = f1;
    std::cout << "-";
}
```

What might happen if you try to compile, link and run this code?

```
#include <iostream>

struct Foo {
    Foo() { std::cout << "a"; }
    Foo(int i) { std::cout << i; }
    ~Foo() { std::cout << "c"; }
};

int main() {
    Foo f1[3];
    std::cout << "-";
    Foo f2[3] = {1,2};
    std::cout << "-";
    Foo * f3 = new Foo[3];
    std::cout << "-";
    delete f3;
    std::cout << "-";
}
```

What might happen if you try to compile, link and run this code?

```cpp
#include <iostream>
#include <string>
#include <algorithm>

size_t count(char ch, std::string const & str)
{
    size_t n = 0;
    char const * p = str.c_str();
    while (n += *p == ch, *p++ != 0)
        ;
    return n;
}


int main()
{
    char const c[] = "Tamatawhakatangih\0angakoauaoutamateapolaiwhenuakitanaahu";
    std::string str(c, sizeof(c));

    if (size_t n = count('w', str)) {
        std::cout << "Character found " << n << " times" << std::endl;
    } else {
        std::cout << "Character not found" << std::endl;
    }

    if (size_t n = std::count(str.begin(), str.end(), 'w')) {
        std::cout << "Character found " << n << " times" << std::endl;
    } else {
        std::cout << "Character not found" << std::endl;
    }
}
```

What might happen if you try to compile, link and run this code?