```
 1 #include <iostream>
 2
 3 static int a = 0;
 4 int b = 0;
 5
 6 void f(int n) {
 7     static int c = 0;
 8     while ( n-- ) {
 9         static int d = 0;
10         int e = 0;
11         std::cout << a++ << b++ << c++ << d++ << e++ << std::endl;
12     }
13 }
14
15 int main() {
16     f(3);
17     f(3);
18 }
```

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

```
 1 #include <iostream>
 2
 3 void f( int a, int & b, int * c ) {
 4     ++a;
 5     ++b;
 6     ++c;
 7 }
 8
 9 int main() {
10     int a = 3;
11     int b = 4;
12     int c = 5;
13     std::cout << a << ',' << b << ',' << c << std::endl;
14     f(a,b,c);
15     std::cout << a << ',' << b << ',' << c << std::endl;
16 }
```

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

```
 1 #include <iostream>
 2
 3 void p(int i) { std::cout << i; }
 4 void p(char ch) { std::cout << ch; }
 5
 6 class A {
 7 public:
 8     A() { p(1); }
 9     A(const A & a) { p(2); }
10     void operator=(const A & a) { p(3); }
11     A(int i) { p(4); }
12     ~A() { p(5); }
13 };
14
15 void f(A & a) { p('f'); }
16 void g(const A & a) { p('g'); }
17 A h() { p('h'); return 1; }
18 const A & i() { p('i'); return 1; }
19
20 int main() {
21     A a;
22     p('-'); f(a);
23     p('-'); g(a);
24     p('-'); g(1);
25     p('-'); a = h();
26     p('-'); a = i();
27     p('-');
28 }
```

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

```
 1 #include <iostream>
 2 #include <string>
 3
 4 void p(double) { std::cout << "p(double)" << std::endl; }
 5 void p(float) { std::cout << "p(float)" << std::endl; }
 6 void p(std::string) { std::cout << "p(std::string)" << std::endl; }
 7 void p(const char *) { std::cout << "p(const char *)" << std::endl; }
 8 void p(char *) { std::cout << "p(char *)" << std::endl; }
 9 void p(long) { std::cout << "p(long)" << std::endl; }
10 void p(char) { std::cout << "p(char)" << std::endl; }
11
12 int main() {
13     p(1.9);
14     p('s');
15     p("hello");
16     p(1);
17 }
```

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

```
 1 #include <iostream>
 2
 3 #define MIN(a,b) (((a)<(b))?(a):(b))
 4
 5 int main() {
 6     int a=2;
 7     int b=3;
 8     int c=0;
 9     std::cout << c << a << b << std::endl;
10     c = MIN(++a,++b);
11     std::cout << c << a << b << std::endl;
12     c = MIN(++a,++b);
13     std::cout << c << a << b << std::endl;
14 }
```

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

```
 1 #include <iostream>
 2
 3 #define MAX(a,b) a > b ? a : b
 4
 5 int main() {
 6     int a=2;
 7     int b=3;
 8     int c=0;
 9     std::cout << c << a << b << std::endl;
10     c = 42 + MAX(a,b);
11     std::cout << c << a << b << std::endl;
12 }
```

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

```
 1 #include <iostream>
 2
 3 int a = 4;
 4
 5 namespace {
 6     int b = 5;
 7 };
 8
 9 int main() {
10     std::cout << a << std::endl;
11     std::cout << b << std::endl;
12 }
```

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

```
 1 #include <iostream>
 2
 3 template<typename T> void p(T x) { std::cout << x; }
 4
 5 void f() { p(1); }
 6
 7 namespace A {
 8     void f() { p(2); }
 9 }
10
11 void g() { p(3); }
12
13 namespace {
14     void g() { p(4); }
15 };
16
17 int main() {
18     f();
19     g();
20 }
```

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

foo.hpp

```
1  namespace Foo {
2      void f();
3  }
```

foo.cpp

```
1  #include <iostream>
2  #include "foo.hpp"
3
4  namespace Foo {
5      void f() { std::cout << "f()" << std::endl; }
6  }
7
8  namespace {
9      void g() { Foo::f(); }
10 }
11
12 int main() {
13     g();
14 }
```

What might happen if you try to compile, link and run this program? Please criticize.

```
 1 #include <iostream>
 2
 3 template<typename T> void p(T x) { std::cout << x; }
 4
 5 struct A {};
 6 struct B : A {};
 7 struct C : B {};
 8 struct D : C {};
 9
10 int main() {
11     try {
12         p(1);
13         throw B();
14         p(2);
15     } catch(A a) {
16         p(3);
17         throw C();
18         p(4);
19     } catch(B & b) {
20         p(5);
21         throw D();
22         p(6);
23     } catch(const C & c) {
24         p(7);
25         throw;
26         p(8);
27     } catch(...) {
28         p(9);
29     }
30     p(0);
31 }
```

What might happen if you try to compile, link and run this program? Criticize the code.

foo.hpp

```
1 int a;
2 const int b = 42;
3 int foo();
```

foo.cpp

```
1 #include "foo.hpp"
2
3 int a = 42;
4 int foo() {
5     return a;
6 }
```

bar.cpp

```
1 #include "foo.hpp"
2 #include <iostream>
3
4 int main() {
5     std::cout << foo() << " is " << b << std::endl;
6 }
```

What might happen if you try to compile, link and run this program using this command line:

```
g++ -Wall foo.cpp bar.cpp && ./a.out
```

foo.hpp

```
1 struct S {
2     char a;
3     int b;
4 };
5
6 S foo();
```

foo.cpp

```
 1 struct S {
 2     int a;
 3     char b;
 4 };
 5
 6 S foo() {
 7     S s;
 8     s.a = 64;
 9     s.b = 'A';
10     return s;
11 }
```

bar.cpp

```
1 #include <iostream>
2 #include "foo.hpp"
3
4 S foo();
5
6 int main() {
7     S s = foo();
8     std::cout << s.a << s.b << std::endl;
9 }
```

What might happen if you try to compile, link and run this program using this command line:

```
g++ -Wall foo.cpp bar.cpp && ./a.out
```

```
 1 #include <iostream>
 2 #include <ctype.h>
 3
 4 struct X {
 5     char id;
 6     X(char ch) : id(ch) { std::cout << (char)toupper(id); }
 7     ~X() { std::cout << id; }
 8 };
 9
10 struct A : X { A() : X('a') {} };
11 struct B : X { B() : X('b') {} };
12 struct C : X { C() : X('c') {} };
13 struct D : X { D() : X('d') {} };
14 struct E : X { E() : X('e') {} };
15 struct F : X { F() : X('f') {} };
16
17 A a;
18 static B b;
19
20 int main() {
21     C c;
22     static D d;
23     {
24         static E e;
25         F f;
26     }
27     return 0;
28 }
```

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