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

int main() {
    std::string s("kulturuke");
    std::cout << s << std::endl;
    while( next_permutation(s.begin(),s.end()) )
        std::cout << s << std::endl;
}
```

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

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

int main() {
    std::string s("Hello World!");
    std::wstring ws(s);
    std::cout << ws << std::endl;
}
```

This code does not compile. How to fix?

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

int main() {
    std::string s1 = "Foo";
    std::string s2 = "Gaz";
    s2 = s1;
    s2[0] = ’B’;
    std::cout << s1 << std::endl;
}
```

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

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

int main() {
    std::string s1 = "abbcccde";
    std::string::size_type p = s1.rfind("cc");
    s1.replace(p, 2, "XXX");
    std::string s2 = s1.substr(3, -2);
    std::cout << s2 << std::endl;
}
```

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

```cpp
#include <iostream>

int main() {
    int a = 4;
    int b = 2;
    std::clog << a << b;
}
```

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

```
#include <iostream>

struct A {
    virtual std::ostream & put(std::ostream &) const = 0;
};

struct B : A {
    std::ostream & put(std::ostream & s) const { return s << ’B’; }
};

std::ostream & operator<<(std::ostream & s, const A & a) {
    return a.put(s);
}

int main() {
    B b;
    std::cout << b << std::endl;
}
```

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

```cpp
#include <iostream>

int main() {
    double pi = 3.14159265358979323846;
    std::cout << pi << std::endl;
    std::cout.precision(3);
    std::cout << pi << std::endl;
}
```

What does this print out? How could this code might look like if we were using a stream manipulator instead?

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

int main() {

    std::string ostr = "This is a test of writing and reading from files";
    std::ofstream ofile("myfile.tmp");
    ofile << ostr;

    std::string istr;
    std::ifstream ifile("myfile.tmp");
    ifile >> istr;

    std::cout << istr;
}
```

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

```
#include <iostream>
#include <limits>

int main() {

    std::cout << std::numeric_limits<char>::digits << std::endl;

    std::cout << std::numeric_limits<int>::digits << std::endl;
    std::cout << std::numeric_limits<int>::max() << std::endl;
    std::cout << std::numeric_limits<int>::min() << std::endl;
}
```

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

```
#include <iostream>
#include <valarray>

double f(double d) {
    return d + 1;
}

int main() {
    const double a[] = {1.23, -4.54, 0.48, -1};
    const double b[] = {1, 0, 0, 1};

    std::valarray<double> va(a,4);
    std::valarray<double> vb(b,4);
    std::valarray<double> vc = va * vb;
    vc *= 2;
    std::valarray<double> vd = vc.apply(f);

    for (size_t i=-0; i<vd.size(); i++)
        std::cout << vd[i] << " ";
}
```

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