Dateien analysieren

```
@Def(file: ana.cpp)
    @put(main prereqs);
    int main(
        int argc, const char *argv[]
    ) {
        @Put(parse args);
        @put(read input);
        @put(write table);
    }
@End(file: ana.cpp)
```

Datenstruktur für Statistik

```
@Def(def collection)
    using Collection =
        std::map<char, int>;
@End(def collection)
```

```
@def(main prereqs)
    #include <map>
    @Put(def collection);
    Collection collection;
@end(main prereqs)
```

```
@add(main prereqs)
    #include <iostream>
@end(main prereqs)
```

```
@def(read input)
    @Put(init state);
    char ch;
    while (std::cin.get(ch)) {
        @Put(add to collection);
    }
@end(read input)
```

```
@Def(add to collection)
    ++collection[ch];
@End(add to collection)
```

```
@def(write table)
    for (const auto &e : collection) {
        @Put(write key);
        std::cout << "\t" <<
            e.second << "\n";
    }
@end(write table)
```

```
@add(main prereqs)
    #include <cctype>
    void write_byte(char b) {
        if (isprint(b) &&
            b != '%' && b > ' '
        ) {
            std::cout << b;
        } else {
            @put(write escaped);
        }
    }
@end(main prereqs)
```

```
@def(write escaped)
    static const char digits[] =
        "0123456789abcdef";
    std::cout << '%' <<
        digits[(b >> 4) & 0xf] <<
        digits[b & 0xf];
@end(write escaped)
```

```
@Def(write key)
    write_byte(e.first);
@End(write key)
```