# Byte-Folgen analysieren

## Byte-Folgen analysieren

- passt das Programm an, um ganze Byte-Folgen fester Länge zu analysieren
- anstatt nur einzelner Bytes

- löscht das Fragment, da es sonst Zwischenstände gibt, die nicht kompilieren
- nach der Typ-Änderung werden die Fragmente neu gefüllt

```
@Rep(write key)
@End(write key)
```

```
@Rep(add to collection)
@End(add to collection)
```

- löscht das Fragment, da es sonst Zwischenstände gibt, die nicht kompilieren
- nach der Typ-Änderung werden die Fragmente neu gefüllt

- definiert `Prefix`

```
@inc(prefix.md)
```

```
@Rep(def collection)
    @Mul(prefix);
    using Collection =
        std::map<Prefix, int>;
@End(def collection)
```

- Collection zählt nun `Prefix` Instanzen

- Zustand ist nun eine `Prefix` Instanz
- und die wird initialisiert

```
@Def(init state)
    Prefix state;
    init(state);
@End(init state)
```

```
@Rep(add to collection)
    push(state, ch);
    ++collection[state];
@End(add to collection)
```

- passt Schlüssel an
- und zählt neuen Schlüssel

- gibt alle Bytes des Schlüssels aus

```
@Rep(write key)
    unsigned i { 0 };
    for (; i < prefix_length; ++i) {
        write_byte(e.first[i]);
    }
@End(write key)
```

## Andere Längen der Byte-Folgen

### Andere Längen der Byte-Folgen

- als Vorgabe werden Byte-Folgen der Länge 2 betrachtet
- um andere Längen zu verwenden, kann die neue Länge mit der Option `-n` auf der Kommandozeile angegeben werden
- so betrachtet zum Beispiel `-n3` Byte-Folgen der Länge 3

- bearbeite Option `-n`

```
@Def(parse args)
    if (argc == 2) {
        const char *arg { argv[1] };
        if (
            arg[0] == '-' &&
                arg[1] == 'n'
        ) {
            @put(change length);
        }
    }
@End(parse args)
```

```
@def(change length)
    prefix_length = std::stoi(arg + 2);
    if (prefix_length < 1) {
        std::cerr << "invalid length\n";
        prefix_length = 2;
    }
@end(change length)
```

- setze neue Länge
- key wird neu instantiiert
- wenn die Länge zu kurz ist, wird statt dessen 2 verwendet