5.26. Реализуйте структуру make_index_sequence<N>, которая бы представляла из себя index_sequence <%последовательность от 0 до N-1%>.

`index_sequence` - в некотором смысле compile-time массив - структура, шаблонными параметрами которой являются size_t, в каком-то смысле она их «хранит» (`integer_sequence` хранит int). Реализуем `make_index_sequence`.

```
template <size_t... Ints>
struct index_sequence {};

template <typename T, size_t N>
struct push_back; // сама структура нам не нужна, нужна только специализация

template <size_t N, size_t... Ints>
struct push_back<index_sequence<Ints...>, N> {
    using type = index_sequence<Ints..., N>;
}; // добавление числа N в конец index_sequence из Ints

template <size_t N>
struct make_index_sequence_s {
    using type = typename push_back<
            typename make_index_sequence_s<N-1>::type,
            N-1>::type; // добавляем N-1 к предыдущему результату
};

template <>
struct make_index_sequence_s<0> {
    using type = index_sequence<>; // база - пустая последовательность
};

template <size_t N>
using make_index_sequence = typename make_index_sequence_s<N>::type;
```

Для проверки того, что все работает, можно запустить следующую строчку:
`static_assert(std::is_same_v< make_index_sequence<3>, index_sequence<0, 1, 2> >);`
Замечание: Пример использования этих структур приведен в билете 5.27.