Week 3

Exercise 18

../18/nrtrait.h

```
1  #ifndef INCLUDED_NRTRAIT_
2  #define INCLUDED_NRTRAIT_
3
4  #include <cstddef>
5  #include <cmath>
6
7  template<size_t n0>
8  struct NrTrait
9  {
10   enum
11   {
12     value = n0,
13     even  = (n0 % 2 == 0),
14     by3   = (n0 % 3 == 0),
15     width = (static_cast<size_t>(std::log10(n0)) + 1)
16   };
17 };
18
19 #endif
```

../18/main.ih

```
1  #define ERR(msg) printf("%s : %d", (msg), __LINE__)
2
3  #include "nrtrait.h"
4
5  #include <iostream>
6
7  using namespace std;
```

../18/main.cc

```
1  #include "main.ih"
2
3  int main(int argc, char const **argv)
4  {
5    constexpr size_t mystNo = 1971962;
6
7    cout << NrTrait<mystNo>::value << '\n'
8         << "Even?           " << NrTrait<mystNo>::even  << '\n'
9         << "Divisible by 3? " << NrTrait<mystNo>::by3   << '\n'
10        << "Width:          " << NrTrait<mystNo>::width << '\n';
11
12 }
```

Exercise 20

../20–2/main.ih

```
1  #define ERR(msg) printf("%s : %d", (msg), __LINE__)
2
3  #include "chars/chars.h"
4  #include "onechar/onechar.h"
5  #include "merge/merge.h"
6
7  #include <iostream>
8
9  using namespace std;
```

../20–2/main.cc

```
1  #include "main.ih"
2
3  int main(int argc, char const **argv)
4  {
5    cout <<
6      Merge<
7          Chars<'1', '2', '3'>,
8          Merge<Chars<'4', '5'>, OneChar<'6'>>::CP
9        >::CP() << '\n';
10 }
```

../20–2/chars/chars.h

```
1  #ifndef INCLUDED_CHARS_
2  #define INCLUDED_CHARS_
3
4  #include <iostream>
5
6  template <char ...CharsT>
7  class Chars
8  {};
9
10 template <char First, char... CharsT>
11 class Chars<First, CharsT...>
12 {
13   public:
14
15     static std::string letters()
16     {
17       return First + Chars<CharsT...>::letters();
18     }
19
20     friend std::ostream &operator<<(std::ostream &out,
21                                     Chars<First, CharsT...> const &rhs)
22     {
23         out << rhs.letters();
24       return out;
25     }
26 };
27
28
29 template <>
30 class Chars<>
31 {
32   public:
33
34     static std::string letters()
35     {
36       return "";
```

```
37        }
38  };
39
40
41  #endif
```

../20–2/merge/merge.h

```
1  #ifndef  INCLUDED_MERGE_
2  #define  INCLUDED_MERGE_
3
4  template <class CharsT1, class CharsT2>
5  class Merge
6  {};
7
8  template <char ...CharsT1, char CharT2>
9  class Merge <Chars<CharsT1...>, OneChar<CharT2>>
10  {
11    public:
12
13      typedef Chars<CharsT1..., CharT2> CP;
14
15  };
16
17  template <char ...CharsT1, char ...CharsT2>
18  class Merge <Chars<CharsT1...>, Chars<CharsT2...>>
19  {
20    public:
21
22      typedef Chars<CharsT1..., CharsT2...> CP;
23  };
24
25
26
27  #endif
```

../20–2/onechar/onechar.h

```
1  #ifndef  INCLUDED_ONECHAR_
2  #define  INCLUDED_ONECHAR_
3
4  template<char character>
5  class OneChar
6  {};
7
8  #endif
```