C++ Exercises
Set1

Author(s): Tino Alferink, Robert van Ark, Owen Givlin
Previously rated by Frank
15:54

September 17, 2024

4

To transform multiple NTBSs into one RSL and provide a rule
when a delimiter needs to be used.

Listing 1: `rsl.cc`

```
#include <iostream>

char const RSL[] =                        // define the RSL identical to the output
R"RSL(
^\s+Encryption key:(\w+)
^\s+Quality=(\d+)
^\s+E?SSID:"([[:print:]]+)"
^\s+ssid="([[:print:]]+)"
)RSL";

int main()
{
    std::cout << RSL << '\n';              // display output to the stream
}
```

```
Output:
^\s+Encryption key:(\w+)
^\s+Quality=(\d+)
^\s+E?SSID:"([[:print:]]+)"
^\s+ssid="([[:print:]]+)"
```

The R"(...)" form cannot be used when dealing with conflicting or nested
delimiters, such as in this case. Therefore we use the R"RSL(...)RSL" form to
avoid conflict within the string.

7

To write a program that initialises a size_t value field
using a bit-fields struct and a union.

Listing 2: `bitfield.cc`

```
#include <iostream>

using namespace std;

struct BitField                           // Declares a bit-field struct that
{                                         // adheres to the given constraints.
    size_t bit0:   1;
    size_t field0: 3;
    size_t field1: 4;
    size_t field2: 5;
    size_t field3: 4;
    size_t field4: 4;
    size_t field5: 4;
    size_t field6: 4;
    size_t field7: 4;
    size_t field8: 5;
    size_t field9: 2;
};

union ValueUnion
{
    size_t value;
    BitField bitField;
};

constexpr ValueUnion valueUnion = { // Set the bitField values at compile time.
    .bitField = {0, 7, 15, 10, 6, 7, 15, 15, 7, 0, 3}
};

int main()
{                                   // Display value in hex.
    cout << hex << valueUnion.value << '\n';
}
```

```
Output:
c0ffeecafe
```