```
// $Id: template_string.cpp,v 1.4 2016-08-02 17:51:49-07 - - $

#include <iostream>
using namespace std;

template <typename char_type>
struct template_string {

   struct long_str {
      size_t capacity_;
      size_t size_;
      char_type* data_;
   };

   static constexpr size_t char_count = (sizeof (long_str) - 1)
                                       / sizeof (char_type);
   static constexpr size_t min_cap = char_count > 2 ? char_count : 2;

   struct short_str {
      union {
         unsigned char size_;
         char_type padding_;
      };
      char_type data_[min_cap];
   };

   union str_rep {
      long_str long_;
      short_str short_;
   } rep_;

   bool is_long_str() const {
      return rep_.short_.size_ & 1;
   }

};

```

```

int offset (const void* base, const void* field) {
   return static_cast<const char*> (field)
        - static_cast<const char*> (base);
}

#define SHOW(X) cout << #X << " = " << X << endl;
template <typename char_type_t>
void print_info (const string& char_type) {
   template_string<char_type_t> v;
   SHOW (char_type);
   SHOW (sizeof (char_type_t));
   SHOW (sizeof v);
   SHOW (sizeof v.rep_);
   SHOW (sizeof v.rep_.long_);
   SHOW (sizeof v.rep_.long_.capacity_);
   SHOW (sizeof v.rep_.long_.size_);
   SHOW (sizeof v.rep_.long_.data_);
   SHOW (sizeof v.rep_.short_);
   SHOW (sizeof v.rep_.short_.size_);
   SHOW (sizeof v.rep_.short_.padding_);
   SHOW (sizeof v.rep_.short_.data_);
   SHOW (v.char_count);
   SHOW (v.min_cap);
   SHOW (offset (&v, &v));
   SHOW (offset (&v, &v.rep_));
   SHOW (offset (&v, &v.rep_.long_));
   SHOW (offset (&v, &v.rep_.long_.capacity_));
   SHOW (offset (&v, &v.rep_.long_.size_));
   SHOW (offset (&v, &v.rep_.long_.data_));
   SHOW (offset (&v, &v.rep_.short_));
   SHOW (offset (&v, &v.rep_.short_.size_));
   SHOW (offset (&v, &v.rep_.short_.padding_));
   SHOW (offset (&v, &v.rep_.short_.data_));
}

#define PRINT(C) print_info<C>(#C);
int main() {
   PRINT(char);
   cout << "\f" << endl;
   PRINT(wchar_t);
   cout << "\f" << endl;
   PRINT(char16_t);
   cout << "\f" << endl;
   PRINT(char32_t);
}

/*
//TEST// ./template_string >template_string.out
//TEST// mkpspdf template_string.ps template_string.cpp* \
//TEST//         template_string.out
*/

```

```
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: starting template_string.cpp
template_string.cpp: $Id: template_string.cpp,v 1.4 2016-08-02 17:51:49-07 - - $
g++ -g -O0 -std=gnu++14 -rdynamic -Wall -Wextra -Wold-style-cast template_string.cpp -o template_string -lglut -lGLU -lGL -lX11 -lrt -lm
cpplint.py.perl template_string.cpp
Done processing template_string.cpp
@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ mkc: finished template_string.cpp
```

```
char_type = char
sizeof (char_type_t) = 1
sizeof v = 24
sizeof v.rep_ = 24
sizeof v.rep_.long_ = 24
sizeof v.rep_.long_.capacity_ = 8
sizeof v.rep_.long_.size_ = 8
sizeof v.rep_.long_.data_ = 8
sizeof v.rep_.short_ = 24
sizeof v.rep_.short_.size_ = 1
sizeof v.rep_.short_.padding_ = 1
sizeof v.rep_.short_.data_ = 23
v.char_count = 23
v.min_cap = 23
offset (&v, &v) = 0
offset (&v, &v.rep_) = 0
offset (&v, &v.rep_.long_) = 0
offset (&v, &v.rep_.long_.capacity_) = 0
offset (&v, &v.rep_.long_.size_) = 8
offset (&v, &v.rep_.long_.data_) = 16
offset (&v, &v.rep_.short_) = 0
offset (&v, &v.rep_.short_.size_) = 0
offset (&v, &v.rep_.short_.padding_) = 0
offset (&v, &v.rep_.short_.data_) = 1
```

```

char_type = wchar_t
sizeof (char_type_t) = 4
sizeof v = 24
sizeof v.rep_ = 24
sizeof v.rep_.long_ = 24
sizeof v.rep_.long_.capacity_ = 8
sizeof v.rep_.long_.size_ = 8
sizeof v.rep_.long_.data_ = 8
sizeof v.rep_.short_ = 24
sizeof v.rep_.short_.size_ = 1
sizeof v.rep_.short_.padding_ = 4
sizeof v.rep_.short_.data_ = 20
v.char_count = 5
v.min_cap = 5
offset (&v, &v) = 0
offset (&v, &v.rep_) = 0
offset (&v, &v.rep_.long_) = 0
offset (&v, &v.rep_.long_.capacity_) = 0
offset (&v, &v.rep_.long_.size_) = 8
offset (&v, &v.rep_.long_.data_) = 16
offset (&v, &v.rep_.short_) = 0
offset (&v, &v.rep_.short_.size_) = 0
offset (&v, &v.rep_.short_.padding_) = 0
offset (&v, &v.rep_.short_.data_) = 4
```

```

char_type = char16_t
sizeof (char_type_t) = 2
sizeof v = 24
sizeof v.rep_ = 24
sizeof v.rep_.long_ = 24
sizeof v.rep_.long_.capacity_ = 8
sizeof v.rep_.long_.size_ = 8
sizeof v.rep_.long_.data_ = 8
sizeof v.rep_.short_ = 24
sizeof v.rep_.short_.size_ = 1
sizeof v.rep_.short_.padding_ = 2
sizeof v.rep_.short_.data_ = 22
v.char_count = 11
v.min_cap = 11
offset (&v, &v) = 0
offset (&v, &v.rep_) = 0
offset (&v, &v.rep_.long_) = 0
offset (&v, &v.rep_.long_.capacity_) = 0
offset (&v, &v.rep_.long_.size_) = 8
offset (&v, &v.rep_.long_.data_) = 16
offset (&v, &v.rep_.short_) = 0
offset (&v, &v.rep_.short_.size_) = 0
offset (&v, &v.rep_.short_.padding_) = 0
offset (&v, &v.rep_.short_.data_) = 2
```

```

char_type = char32_t
sizeof (char_type_t) = 4
sizeof v = 24
sizeof v.rep_ = 24
sizeof v.rep_.long_ = 24
sizeof v.rep_.long_.capacity_ = 8
sizeof v.rep_.long_.size_ = 8
sizeof v.rep_.long_.data_ = 8
sizeof v.rep_.short_ = 24
sizeof v.rep_.short_.size_ = 1
sizeof v.rep_.short_.padding_ = 4
sizeof v.rep_.short_.data_ = 20
v.char_count = 5
v.min_cap = 5
offset (&v, &v) = 0
offset (&v, &v.rep_) = 0
offset (&v, &v.rep_.long_) = 0
offset (&v, &v.rep_.long_.capacity_) = 0
offset (&v, &v.rep_.long_.size_) = 8
offset (&v, &v.rep_.long_.data_) = 16
offset (&v, &v.rep_.short_) = 0
offset (&v, &v.rep_.short_.size_) = 0
offset (&v, &v.rep_.short_.padding_) = 0
offset (&v, &v.rep_.short_.data_) = 4
```