
Subject: Re: get_i

Posted by [Novo](#) on Wed, 17 Jun 2020 04:39:56 GMT

[View Forum Message](#) <> [Reply to Message](#)

Novo wrote on Tue, 16 June 2020 15:42mirek wrote on Tue, 16 June 2020 15:25Novo wrote on Tue, 16 June 2020 21:22Another implementation using initializer_list:

```
template <typename T>
constexpr T get_i(int i, std::initializer_list<T> list)
{
    return list[clamp(i, 0, list.size())];
}
```

This one is trictly-typed, although I couldn't check assembly with godbolt because it complains about something ...

Nah, we do not want strict typing here.

Sorry, last one won't compile.

The one using variadic template is fine, although it still needs specialization for const char* ... :-/

Fixed version. No performance degradation.

```
template <typename T>
constexpr T get_i2(int i, const std::initializer_list<T>& list)
{
    const int n = list.size();
    return *(list.begin() + clamp(i, 0, n));
}
```

```
const char* c = get_i2(1, {"zero", "one", "two"});
```

Assembler:

```
.L.str:
```

```
    .asciz "one"
```

```
c:
```

```
    .quad .L.str
```
