UpperCaseString.cpp

```cpp
// COS30008, Midterm, 2020

#include "UpperCaseString.h"

#include <cctype>
#include <stdexcept>

using namespace std;

UpperCaseString::UpperCaseString( const char* aInitialValue )
{
    // a C-string contains at least one character: '\0'.
    size_t lLength = 1;

    // compute the length + 1 of aInitialValue
    for ( int i = 0; aInitialValue[i] != '\0'; i++ )
    {
        lLength++;
    }

    // allocate memory
    fValue = new char[lLength];

    // copy aInitialValue to fValue and make it upper case
    for ( int i = 0; i < lLength; i++ )
    {
        fValue[i] = toupper( aInitialValue[i] );
    }
}

UpperCaseString::~UpperCaseString()
{
    delete [] fValue;
}

int UpperCaseString::size() const
{
    size_t lLength = 0;

    // compute the length + 1 of aInitialValue
    for (int i = 0; fValue[i] != '\0'; i++)
    {
        lLength++;
    }

    return lLength;
}

const char UpperCaseString::operator[](int aIndex) const
{
    return fValue[aIndex];
}

UpperCaseStringIterator UpperCaseString::begin() const
{
    UpperCaseStringIterator result(*this, 0);
    return result;
}
```

```
UpperCaseStringIterator UpperCaseString::end() const
{
    UpperCaseStringIterator result(*this, size());
    return result;
}

UpperCaseStringIterator UpperCaseString::rbegin() const
{
    UpperCaseStringIterator result(*this, size()-1);
    return result;
}

UpperCaseStringIterator UpperCaseString::rend() const
{
    UpperCaseStringIterator result(*this, -1);
    return result;
}

std::ostream& operator<<(std::ostream& aOStream, const UpperCaseString& aString)
{
    aOStream << aString.fValue;
    return aOStream;
}
```

UpperCaseStringIterator.cpp

```
#include "UpperCaseStringIterator.h"
#include "UpperCaseString.h"
using namespace std;

UpperCaseStringIterator::UpperCaseStringIterator(const UpperCaseString& aString, int
aStart) : fString(aString), fIndex(aStart)
{
}

const char UpperCaseStringIterator::operator*() const
{
	return fString[fIndex];
}

UpperCaseStringIterator UpperCaseStringIterator::operator++(int)
{
	UpperCaseStringIterator previous = *this;
	this->fIndex++;
	return previous;
}

UpperCaseStringIterator UpperCaseStringIterator::operator--()
{
	UpperCaseStringIterator previous = *this;
	this->fIndex--;
	return previous;
}

bool UpperCaseStringIterator::operator==(const UpperCaseStringIterator& aOther) const
{
	return (this->fIndex == aOther.fIndex);
}

bool UpperCaseStringIterator::operator!=(const UpperCaseStringIterator& aOther) const
{
	return !(*this == aOther);
}

UpperCaseStringIterator UpperCaseStringIterator::begin() const
{
	UpperCaseStringIterator result = *this;
	result.fIndex = 0;
	return result;
}

UpperCaseStringIterator UpperCaseStringIterator::end() const
{
	UpperCaseStringIterator result = *this;
	result.fIndex = result.fString.size();
	return result;
}

UpperCaseStringIterator UpperCaseStringIterator::rbegin() const
{
	UpperCaseStringIterator result = *this;
	result.fIndex = -1;
	return result;
}

UpperCaseStringIterator UpperCaseStringIterator::rend() const
```

```
{
        UpperCaseStringIterator result = *this;
        result.fIndex = result.fString.size() - 1;
        return result;
}
```

Main.cpp

```
// COS30008, Midterm, 2020

#include "UpperCaseString.h"
#include "UpperCaseStringIterator.h"

using namespace std;

int main()
{
	UpperCaseString s( "Able was I I saw Elba" );

	cout << "The string: \"" << s << "\" with size: " << s.size() << endl;

    bool lPalindrome = true;

    UpperCaseStringIterator move_to_right = s.begin();
    UpperCaseStringIterator move_to_left = s.rbegin();
    UpperCaseStringIterator past_left = s.rend();

	while (move_to_left != past_left)
    {
        if (*move_to_right != *move_to_left) {
            lPalindrome = false;
            break;
        }

        move_to_right++;
        --move_to_left;
    }

	if ( lPalindrome )
		cout << "We have found a palindrome!" << endl;
	else
		cout << "Oops, no palindrome." << endl;

	return 0;
}
```

OUTPUT:

```
The string: "ABLE WAS I I SAW ELBA" with size: 21
We have found a palindrome!

C:\Users\somem\OneDrive - Swinburne University\DSAP\Assignment\Midterm_Test\program\midtermprogram\Debug\midtermprogram.
exe (process 3584) exited with code 0.
To automatically close the console when debugging stops, enable Tools->Options->Debugging->Automatically close the conso
le when debugging stops.
Press any key to close this window . . .
```