CS205 C/C++ Programming - Lab ASSIGNMENT 5

Name : 杨睦圳（Yang Muzhen）

SID ： 11813010

Part 1 - Analysis

We can take use of the member function of `std::string` and the `utf8.c` library

`length()`: we can use the function `utf8_charlen()` in `utf8.c` library.

`bytes()`: we can use the function `length()` in `std::string`

`find()` : we can subtract the length of origin string by the length of the substring after the target

`replace()`: we can use a while loop with a the function `replace()` in the `std::string`

`<<`: realize it by the way we learn in class

`+`: add the two string member and return a new object

`+=`: add the string member of right object to the string member of the left object and return the left object

`*`: add the string multiple time and return a new object

`!`: use the similar way of the function `utf8_charlen()` in `utf8.c` library, get all the characters and concat then reversely in a string.

Part 2 - Code

The program is worked on cygwin

UTF8string.hpp

```
#ifndef LAB101_UTF8STRING_HPP
#define LAB101_UTF8STRING_HPP
#include <cstring>
#include <iostream>
using namespace std;
class UTF8string {
    friend ostream& operator<<(ostream&,const UTF8string&);
    friend UTF8string operator*(int, const UTF8string&);
    friend UTF8string operator!(const UTF8string&);

private:
    string s;
public:
    UTF8string(string);
    UTF8string(const char*);

    int length();
    int bytes();
    int find(string substr);
    void replace(UTF8string to_remove, UTF8string replacement);
    UTF8string operator+(const UTF8string&);
    UTF8string operator+=(const UTF8string&);
    UTF8string operator*(int);
};


#endif //LAB101_UTF8STRING_HPP
```

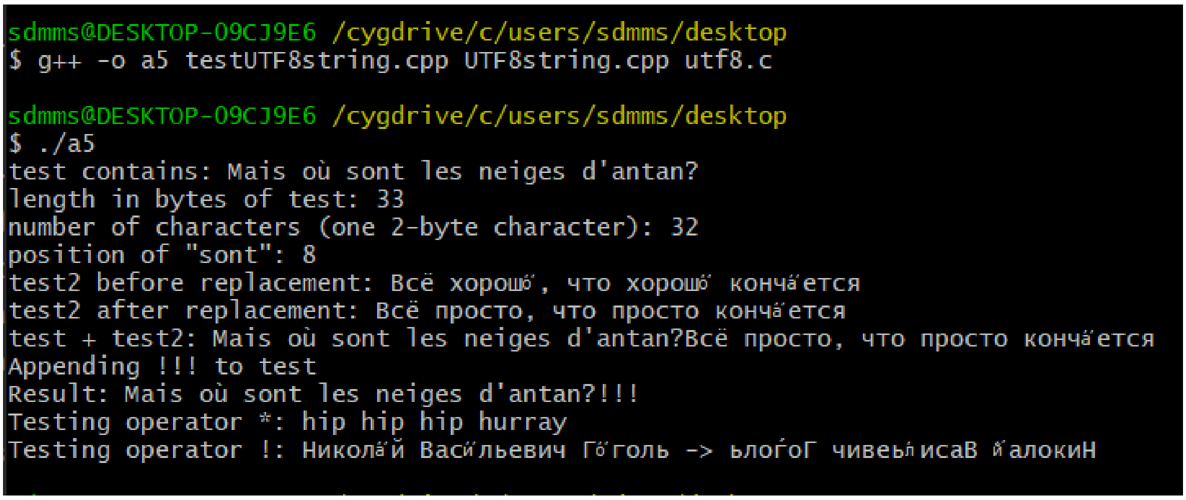
UTF8string.cpp

```
//
// Created by sdmms on 2020/5/3.
//

#include "UTF8string.hpp"
#include "utf8.h"

using namespace std;
UTF8string::UTF8string(string s) {
    this->s = move(s);
}

UTF8string::UTF8string(const char* s) {
    this->s = s;
}

int UTF8string::length() {
    auto* p = (unsigned char*)s.c_str();
    return utf8_charlen(p);
}

int UTF8string::bytes() {
    return s.length();
}

int UTF8string::find(string substr) {
    auto* p = (unsigned char*)s.c_str();
    auto* q = (unsigned char*)substr.c_str();
    auto* pos = utf8_search(p,q);
    return pos == nullptr?
                -1 : utf8_charlen(p) - utf8_charlen(pos);
}

void UTF8string::replace(UTF8string to_remove, UTF8string replacement) {
    int pos;
    while((pos = s.find(to_remove.s)) != string::npos)
        this->s.replace(pos, to_remove.bytes(),replacement.s);
}

ostream& operator<<(ostream& os, const UTF8string& a){
    os << a.s;
    return os;
}
```

```
UTF8string UTF8string::operator+(const UTF8string &a) {
    return UTF8string(s + a.s);
}

UTF8string UTF8string::operator+=(const UTF8string &a) {
    s.append(a.s);
    return *this;
}

UTF8string UTF8string::operator*(int t) {
    int i;
    string str;
    for(i = 0; i < t; i++)
        str.append(s);
    return UTF8string(str);
}

UTF8string operator*(const int t, const UTF8string& a){
    int i;
    string s;
    for(i = 0; i < t; i++)
        s.append(a.s);
    return UTF8string(s);
}

UTF8string operator!(const UTF8string& a){
    string str1 = a.s, str2;
    auto* p = (unsigned char*) a.s.c_str();
    int pos = 0,len;

    while(*p != '\0'){
        len = isutf8(p);
        if(len == 0){
            str2 = str1.substr(pos,1) + str2;
            ++pos;
            ++p;
        }else{
            str2 = str1.substr(pos,len) + str2;
            pos += len;
            p += len;
        }
    }

    return UTF8string(str2);
}
```

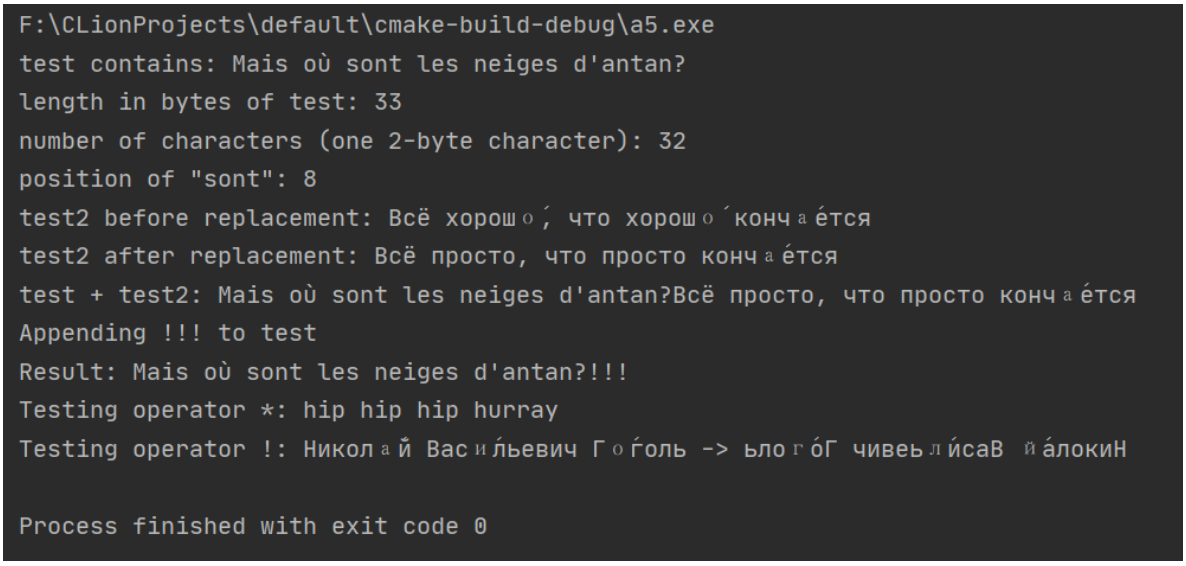
Part 3 - Result & Verification

Compile and Result on cmd:



```
sdmms@DESKTOP-09CJ9E6 /cygdrive/c/users/sdmms/desktop
$ g++ -o a5 testUTF8string.cpp UTF8string.cpp utf8.c

sdmms@DESKTOP-09CJ9E6 /cygdrive/c/users/sdmms/desktop
$ ./a5
test contains: Mais où sont les neiges d'antan?
length in bytes of test: 33
number of characters (one 2-byte character): 32
position of "sont": 8
test2 before replacement: Всё хорошо́, что хорошо́ конча́ется
test2 after replacement: Всё просто, что просто конча́ется
test + test2: Mais où sont les neiges d'antan?Всё просто, что просто конча́ется
Appending !!! to test
Result: Mais où sont les neiges d'antan?!!!
Testing operator *: hip hip hip hurray
Testing operator !: Никола́й Васи́льевич Го́голь -> ьлоѓоГ чивеьл́исаВ й́алокиН
```

Result on clion:



```
F:\CLionProjects\default\cmake-build-debug\a5.exe
test contains: Mais où sont les neiges d'antan?
length in bytes of test: 33
number of characters (one 2-byte character): 32
position of "sont": 8
test2 before replacement: Всё хорошо́, что хорошо́ конча́ется
test2 after replacement: Всё просто, что просто конча́ется
test + test2: Mais où sont les neiges d'antan?Всё просто, что просто конча́ется
Appending !!! to test
Result: Mais où sont les neiges d'antan?!!!
Testing operator *: hip hip hip hurray
Testing operator !: Никола́й Васи́льевич Го́голь -> ьлоѓоГ чивеьл́исаВ й́алокиН

Process finished with exit code 0
```

Part 4 - Difficulties & Solutions

When I first use the `replace()` function, I find that only the first target was replaced. A while loop is needed and we will call the `replace()` function until there is no target.