CS205 C/ C++ Programming Lab Assignment5

> Name: 黄玉安 (Huang Yu'an)
>
> SID: 11610303

Part1 - Analysis

I designed my UTF8string like following, it only has one attribute ori: which is a string and one constructure and serveral member functions as well as two friend functions.

```
class UTF8string{
    string ori;

public:
    UTF8string(const char* str);
    int length();
    int bytes();
    int find(string substr);
    void replace(UTF8string to_remove, UTF8string replacement);

    friend ostream& operator<<(ostream&os, const UTF8string&other);
    UTF8string operator+(UTF8string& another) const;
    void operator+=(UTF8string another);
    UTF8string operator*(int times);
    friend UTF8string operator*(int times, UTF8string& ori);
    UTF8string operator!();

};
```

Part2 -Code

I used some important functions declared in utf8.h:

```
extern          int   utf8_charlen(unsigned char *p);
extern          int   utf8_charpos_to_bytes(unsigned char *s, int pos);
extern unsigned char *utf8_search(const unsigned char *haystack,
                                  const unsigned char *needle);
```

I used `int utf8_charlen(unsigned char *p)` to find the length of my UTF8string, for UTF8string::find(string substr), UTF8string::replace(UTF8string to_remove, UTF8string replacement) and UTF8string::operator!(), it is a little complex. The detaild of them is showed below.

```
int UTF8string::find(string substr){
    unsigned char* res = utf8_search((unsigned char *)ori.c_str(),
                                  (unsigned char *)substr.c_str());
    return length()- utf8_charlen(res);
}
void UTF8string::replace(UTF8string to_remove, UTF8string replacement){
    int position = find(to_remove.ori);
    position = utf8_charpos_to_bytes((unsigned char*)this->ori.c_str(),
position);
    ori.replace(position, to_remove.bytes(), replacement.ori);
}

UTF8string UTF8string::operator!(){
    int len = length();
    unsigned char* cstr = (unsigned char*)ori.c_str();
    int pos = bytes();
    string res = "";
    for(int i=len-1;i>=0;i--){
        int start = utf8_charpos_to_bytes(cstr, i);
        res.append(this->ori.substr(start, pos-start));
        pos = start;
    }
    UTF8string sres(res.c_str());
    return sres;
}
```

You can find more details in my submited code file.

Part 3 - Result & Verification

using make to compile it and using ./main in terminal to run it.

```
➜ assign5 ./main
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 хорошó, что хорошó кончáется
test2 after replacement: Всё просто, что хорошó кончáется
test + test2: Mais où sont les neiges d'antan?Всё просто, что хорошó кончáется
Appending !!! to test
Result: Mais où sont les neiges d'antan?!!!
Testing operator *: hip hip hip hurray
Testing operator !: Никола́й Васи́льевич Го́голь -> ьлоѓоГ чивеьл́исаВ йалокиН
```

The reslut is the same as what we want.

Part 4 - Difficulties & Solutions

It may be confused that we have a function `void replace(UTF8string to_remove, UTF8string replacement);` while we pass the string as argument when it be called. It is the compiler that implicily call the constructor to change to the target object.