```
// String.h

#ifndef STRING_INCLUDED
#define STRING_INCLUDED

class String
{
  public:
    String(const char* value = "");
    String(const String& other);
    ~String();
    String& operator=(const String& rhs);
    void swap(String& other);
    int size() const;
  private:
    char* m_text;
    int m_size;
};

#endif // STRING_INCLUDED

==========================================================

// String.cpp

#include "String.h"
#include <cstring>
using namespace std;

String::String(const char* value)
{
    if (value == nullptr)
        value = "";
    m_size = strlen(value);
    m_text = new char[m_size+1];
    strcpy(m_text, value);
}

String::String(const String& other)
 : m_size(other.m_size)
{
    m_text = new char[m_size+1];
    strcpy(m_text, other.m_text);
}

String::~String()
{
    delete [] m_text;
}

String& String::operator=(const String& rhs)
{
    if (this != &rhs)
    {
        String temp(rhs);
        swap(temp);
    }
    return *this;
}

void String::swap(String& other)
{
    ... // exchange the m_size and other.m_size ints
    ... // exchange the m_text and other.m_text pointers
}

int String::size() const
{
    return m_size;
}
```