1.

#include <iostream>

#include <iomanip>

#include <cstring>

using namespace std ;

class String

{

private:

char \*s ;

public:

String(const char \*a = NULL)

{

int i = strlen(a) ;

s = new char[i+1] ;

strcpy(s, a) ;

cout << "构造：" << s << endl ;

} ;

String(String &a)

{

int i = strlen(a.s) ;

s = new char[i+1] ;

strcpy(s, a.s) ;

cout << "复制构造：" << s << endl ;

}

~String()

{

cout << "析构:" << s << endl ;

}

void Set(const char \*a)

{

int i = strlen(a) ;

s = new char[i+1] ;

strcpy(s, a) ;

}

void Print()

{

cout << s << endl ;

}

bool operator ==(String &s2) ;

bool operator >(String &s2) ;

bool operator <(String &s2) ;

bool operator !=(String &s2) ;

bool operator >=(String &s2) ;

bool operator <=(String &s2) ;

} ;

bool String::operator ==(String &s2)

{

if(strcmp(s, s2.s) == 0)

return 1;

else return 0 ;

}

bool String::operator >(String &s2)

{

if(strcmp(s, s2.s) > 0)

return 1;

else return 0 ;

}

bool String::operator <(String &s2)

{

if(strcmp(s, s2.s) < 0)

return 1;

else return 0 ;

}

bool String::operator !=(String &s2)

{

if(strcmp(s, s2.s) != 0)

return 1;

else return 0 ;

}

bool String::operator >=(String &s2)

{

if(strcmp(s, s2.s) >= 0)

return 1;

else return 0 ;

}

bool String::operator <=(String &s2)

{

if(strcmp(s, s2.s) <= 0)

return 1;

else return 0 ;

}

void main()

{

String s1("abcd"), s2("abce");

if( s1 > s2)

cout << "s1 > s2!\n";

else if ( s1 < s2)

cout << "s1 < s2!\n";

else

cout << "s1 = s2!\n" ;

}

Result:

构造：abcd

构造：abce

s1 < s2!

析构:abce

析构:abcd

请按任意键继续. . .

2.

Code:

#include <iostream>

#include <cstring>

using namespace std ;

class String

{

private:

char \*s ;

public:

String(const char \*a)

{

int n = strlen(a) ;

s = new char[n+1] ;

strcpy(s, a) ;

cout << "构造：" << s <<endl ;

}

String(String &a)

{

int i = strlen(a.s) ;

s = new char[i+1] ;

strcpy(s, a.s) ;

cout << "复制构造：" << s <<endl ;

}

~String()

{

cout << "析构:" << s << endl ;

}

void Set(const char \*a)

{

int i = strlen(a) ;

s = new char[i+1] ;

strcpy(s, a) ;

}

friend int operator >(String s1, String s2) ;

friend int operator <(String s1, String s2) ;

friend int operator >=(String s1, String s2) ;

friend int operator <=(String s1, String s2) ;

friend int operator ==(String s1, String s2) ;

friend int operator !=(String s1, String s2) ;

} ;

int operator >(String s1, String s2)

{

if (strcmp(s1.s, s2.s) > 0)

return 1 ;

else

return 0 ;

}

int operator <(String s1, String s2)

{

if (strcmp(s1.s, s2.s) < 0)

return 1 ;

else

return 0 ;

}

int operator ==(String s1, String s2)

{

if (strcmp(s1.s, s2.s) == 0)

return 1 ;

else

return 0 ;

}

int operator !=(String s1, String s2)

{

if (strcmp(s1.s, s2.s) != 0)

return 1 ;

else

return 0 ;

}

int operator >=(String s1, String s2)

{

if (strcmp(s1.s, s2.s) >= 0)

return 1 ;

else

return 0 ;

}

int operator <=(String s1, String s2)

{

if (strcmp(s1.s, s2.s) <= 0)

return 1 ;

else

return 0 ;

}

void main()

{

String s1("abcd"), s2("abce") ;

if( s1 > s2)

cout << "s1 > s2!\n";

else if ( s1 < s2)

cout << "s1 < s2!\n";

else

cout << "s1 = s2!\n" ;

}

Result:

构造：abcd

构造：abce

复制构造：abce

复制构造：abcd

析构:abcd

析构:abce

复制构造：abce

复制构造：abcd

析构:abcd

析构:abce

s1 < s2!

析构:abce

析构:abcd

请按任意键继续. . .

3.

第二题中，需要调用的两个函数将形参改为引用形式之后：

Code:

#include <iostream>

#include <cstring>

using namespace std ;

class String

{

private:

char \*s ;

public:

String(const char \*a)

{

int n = strlen(a) ;

s = new char[n+1] ;

strcpy(s, a) ;

cout << "构造：" << s <<endl ;

}

String(String &a)

{

int i = strlen(a.s) ;

s = new char[i+1] ;

strcpy(s, a.s) ;

cout << "复制构造：" << s <<endl ;

}

~String()

{

cout << "析构:" << s << endl ;

}

void Set(const char \*a)

{

int i = strlen(a) ;

s = new char[i+1] ;

strcpy(s, a) ;

}

friend int operator >(String &s1, String &s2) ;

friend int operator <(String &s1, String &s2) ;

friend int operator >=(String s1, String s2) ;

friend int operator <=(String s1, String s2) ;

friend int operator ==(String s1, String s2) ;

friend int operator !=(String s1, String s2) ;

} ;

int operator >(String &s1, String &s2)

{

if (strcmp(s1.s, s2.s) > 0)

return 1 ;

else

return 0 ;

}

int operator <(String &s1, String &s2)

{

if (strcmp(s1.s, s2.s) < 0)

return 1 ;

else

return 0 ;

}

int operator ==(String s1, String s2)

{

if (strcmp(s1.s, s2.s) == 0)

return 1 ;

else

return 0 ;

}

int operator !=(String s1, String s2)

{

if (strcmp(s1.s, s2.s) != 0)

return 1 ;

else

return 0 ;

}

int operator >=(String s1, String s2)

{

if (strcmp(s1.s, s2.s) >= 0)

return 1 ;

else

return 0 ;

}

int operator <=(String s1, String s2)

{

if (strcmp(s1.s, s2.s) <= 0)

return 1 ;

else

return 0 ;

}

void main()

{

String s1("abcd"), s2("abce") ;

if( s1 > s2)

cout << "s1 > s2!\n";

else if ( s1 < s2)

cout << "s1 < s2!\n";

else

cout << "s1 = s2!\n" ;

}

Result:

构造：abcd

构造：abce

s1 < s2!

析构:abce

析构:abcd

请按任意键继续. . .