

Шаблон класса My_string

```
#include <string.h>

#include <malloc.h>

class My_string { char* s; int l;

public:

My_string(const char* p1=NULL, const char* p2=NULL) { char* b;

    if(p1 == NULL ) { s = NULL; l = -1; return; }

    else if(p2 == NULL) { p2 = p1; while(*p2++); p2--; }

    l = p2 - p1; b = s = new char[l+1];

    while(p1<p2) *b++ = *p1++; *b='\0'; }

void operator=(const char* x) { delete s; s = new char[(l=strlen(x))+1]; strcpy(s,x); }

void operator=(const My_string& x) { delete s; s = new char[(l=x.l)+1]; strcpy(s,x.s); }

My_string operator+(const My_string& a) {My_string loc;

    loc.l=l+a.l; loc.s = new char [loc.l+1]; strcpy(loc.s,s);

    strcat(loc.s,a.s); return loc; }

char& operator[](int i) { return s[i]; }

bool operator==(const char* x) { for(int i=0;i<=l;i++)

    if(s[i]!=x[i]) return false; return true; }

bool operator==(const My_string& x) { if(l!=x.l) return false;

    for(int i=0;i<l;i++) if(s[i]!=x.s[i]) return false; return true; }

bool operator<(const My_string& x) {
```

Шаблон класса My_string

```
    for(int i=0;i<=l;i++)
if(s[i]==x.s[i]) continue; else return s[i]<x.s[i];

    return false; }

bool operator>(const My_string& x) {
    for(int i=0;i<=l;i++)
if(s[i]==x.s[i]) continue; else return s[i]>x.s[i];

    return false; }

int size() { return l; }

int find(const My_string& x) {
    for(int i=0; i<l; i++) if(!strncmp(s+i,x.s,x.l)) return i;
    return -1; }

~My_string() { delete s; l = -1; }

friend ostream& operator<<(ostream& a, const My_string& x) {
    return a << x.s; }

friend istream& operator>> (istream& b, My_string& a) {
const int n=20; char c,s[n];

    delete a.s; *(a.s = new char)='\0'; a.l=0;
    if(!b.get(c) || c=='\n') return b;
    do { b.putback(c); b.get(s,n); a.s=(char*)realloc(a.s,(a.l+=strlen(s))+1);
strcat(a.s,s); b.get(c); } while(c!='\n');

    return b; };
```