MyStr

Deadline: September 9th, 2014

1 Instructions

You must implement the following functions, whose behavior must match the documentation for the standard function of the same name (without `my_`):

```
const char *my_strchr(const char *s, int c);
unsigned my_strlcpy(char *dst, const char *src, unsigned n);
unsigned my_strlcat(char *dst, const char *src, unsigned n);
int my_strncmp(const char *s1, const char *s2, unsigned n);
void *my_memcpy(void *dst, const void *src, unsigned n);
void *my_memset(void *dst, int c, unsigned n);
```

You can optionally also implement the following for a higher grade (see Grading below):

```
const char *my_strrchr(const char *s, int c);
const char *my_strpbrk(const char *s1, const char *s2);
const char *my_strstr(const char *s1, const char *s2);
void *my_memmove(void *dst, const void *src, unsigned n);
```

Constraints:

- each function must be implemented in a `.c` file of its own, named after the function it contains. The function prototypes must be declared in a `.h` file, in accordance with the C coding standard. The submitted archive may (but needs not) include a test program.
- you must not include any standard/system header in your code; nor use any function from the standard C library.

2 Grading

- 1 point per function correctly implemented in the mandatory list.
- +0.5 if all of the above, and a `Makefile` places the functions in `libminic.a`.
- +0.5 per optional function correctly implemented after all of the above.
- +0.5 if the submitted archive includes a `README` file alongside the source files, containing a valid explanation of why this assignment uses "`const char *`" for `my_strchr/my_strrchr` whereas the C library uses "`char *`" for `strchr/strrchr`.

- +1 point for "cleverness" in either `my_strstr`, `my_memcpy` or `my_memmove`, provided with an explanation and empirical proof of the improvement compared to a naive implementation.

---

3 Copyright and licensing