# Safe String Operations

String operations have serious potential for buffer overflow.
General rules:

- When use strings, check that sufficient memory is allocated before storing any data. In particular, make sure there is space for the `\0` character at the end.
- Make sure all strings are terminated with `\0`.
- All functions which store user input directly as a string are inherently unsafe as the length of the string is not known, and should not be used. Examples are: `scanf "%s"` and `gets`. Suitable replacements are `fgets` or `getline`.

Revised examples:

```
#include <stdio.h>
#include <string.h>

int main () {
    char src[40];
    char dest[100];

    printf ("Enter source string: ");
    fgets(src, 40, stdin);
    strcpy(dest, src);

    printf ("Destination string: %s", dest);

    return 0;
}
```

```
#include <stdio.h>
#include <string.h>
#include <stdlib.h>

int main () {
    char *str = NULL;
    size_t n = 0;
    int res;

    printf ("Enter source string: ");
    res = getline(&str, &n, stdin);
    if (res == -1) {
printf ("Could not read from terminal, exiting\n");
exit(1);
    }
    printf ("Length is %ld\n", strlen(str));
    return 0;

}
```