```c
// Created by hfwei on 2024/11/20.

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

char *StrCat(char *s1, const char *s2);
char *StrCatGLibC(char *dest, const char *src);

char *StrNCat(char *s1, const char *s2, size_t n);
size_t StrNLen(const char *s, size_t max);
char *StrNCatGLic(char *s1, const char *s2, size_t n);

int main(void) {
  char str[50] = "Hello ";
  char str2[50] = "World!";

//  strcat(str, str2);
//  strcat(str, " ...");
//  strcat(str, " Goodbye World!");

//  strcat(str, str2);
//  strncat(str, " Goodbye World!", 3);

  puts(str);

  return 0;
}
```

```c
// Created by hfwei on 2024/11/20.

#include <stdio.h>

int StrCmp(const char *s1, const char *s2);
int StrCmpStd(const char *s1, const char *s2);
int StrCmpGLibC(const char *p1, const char *p2);

int StrNCmpStd(const char *s1, const char *s2, int n);

int main() {
  const char *str1 = "hi, C";
  const char *str2 = "hi, c";

  printf("StrCmp(\"%s\", \"%s\") = %d\n", str1, str2,
StrCmp(str1, str2));

  // printf("StrCmpStd(\"%s\", \"%s\") = %d\n",
  //        str1, str2, StrCmpStd(str1, str2));
  //
  // int n = 2;
  // printf("StrNCmp(\"%s\", \"%s\", %d) = %d\n",
  //        str1, str2, n, StrNCmp(str1, str2, n));

  return 0;
}
```

```c
// Created by hfwei on 2024/11/20.
//
// C Operator Precedence:
// https://en.cppreference.com/w/c/language/operator_precedence#:~:text=C%20Operator%20Precedence%20%20%20Precedence%20,union%20member%20access%20%2028%20more%20rows%20

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

void StrCpy1(char *dest, const char *src);
void StrCpy2(char *dest, const char *src);
void StrCpy3(char *dest, const char *src);
void StrCpy4(char *dest, const char *src);
void StrCpy5(char *dest, const char *src);
void StrCpy6(char *dest, const char *src);

int main() {
  const char *src = "Hello World";
  char *dest = malloc(strlen(src) + 1);

  StrCpy5(dest, src);
  printf("dest = %s\n", dest);

//  strlen(StrCpyStd(dest, src));

  return 0;
}
```

```c
// file: strlen.c
// Created by hfwei on 2024/11/20.
// See https://en.cppreference.com/w/c/string/byte/strlen

#include <stdio.h>

int StrLen1(const char *s);
int StrLen2(const char *s);
int StrLen3(const char *s);

int main() {
  char msg[] = "Hello World!";

//  printf("StrLen(%s) = %d\n", msg, StrLen1(msg));
//  printf("StrLenStd(%s) = %zu\n", msg, StrLenStd(msg));

  return 0;
}
```

```
 1 # `9-more-pointers`
 2
 3 ## `strlen.c`
 4
 5 - C string literal
 6 - `while (str[len++] != '\0')` vs.
 7   `while (++str[len] != '\0')` vs.
 8   `while (++str[len])`
 9 - `\0` vs. `0`
10
11 ## `strcpy.c`
12
13 ## `strcmp.c`
```

```
add_executable(str-literals str-literals.c)

add_executable(strlen strlen.c)
add_executable(strcpy strcpy.c)
add_executable(strcmp strcmp.c)
add_executable(strcat strcat.c)
```

```c
// Created by hfwei on 2024/11/20.
// Visualization:
// https://pythontutor.com/render.html#code=%23include%20%3Cstdio.h%3E%0A%0Aint%20main%28void%29%20%7B%0A%20%20char%20msg%5B%5D%20%3D%20%22Hello%20World!%22%3B%0A%20%20msg%5B0%5D%20%3D%20'N'%3B%0A%20%20printf%28%22%25s%5Cn%22,%20msg%29%3B%0A%0A%20%20char%20*ptr_msg%20%3D%20%22Goodbye%20World!%22%3B%0A%20%20ptr_msg%5B0%5D%20%3D%20'N'%3B%0A%20%20printf%28%22%25s%5Cn%22,%20msg%29%3B%0A%0A%20%20return%200%3B%0A%7D&cumulative=true&curInstr=0&heapPrimitives=nevernest&mode=display&origin=opt-frontend.js&py=c_gcc9.3.0&rawInputLstJSON=%5B%5D&textReferences=false
// See String literals: https://en.cppreference.com/w/c/language/string_literal

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

int main(void) {
  char msg[] = "Hello World!";
  msg[0] = 'N';
  printf("%s\n", msg);

  char *ptr_msg = "Goodbye World!";
  ptr_msg[0] = 'N';
  printf("%s\n", msg);

  return 0;
}
```