```c
//
// Created by hfwei on 2022/11/25.
//

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

/**
 * @brief We assume that there is enough room for storing src.
 *  Otherwise, it is an undefined behavior.
 *
 *  If copying takes place between objects that overlap,
 *  then behavior is undefined.
 *
 * @param dest
 * @param src
 */
void StrCpy(char *dest, const char *src);
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);
char *StrCpyStd(char *dest, const char *src);

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

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

  strlen(StrCpyStd(dest, src));

  return 0;
}

void StrCpy(char *dest, const char *src) {
  int i = 0;
  while (src[i] != '\0') {
    dest[i] = src[i];
    i++;
  }

  dest[i] = '\0';
}

void StrCpy1(char *dest, const char *src) {
  int i = 0;
  while ((dest[i] = src[i]) != '\0') {
    i++;
  }
}
```

```c
54 }
55
56 void StrCpy2(char *dest, const char *src) {
57   int i = 0;
58   // dest[i] : *(dest + i)
59   while ((*(dest + i) = *(src + i)) != '\0') {
60     i++;
61   }
62 }
63
64 void StrCpy3(char *dest, const char *src) {
65   while ((*dest = *src) != '\0') {
66     src++;
67     dest++;
68   }
69 }
70
71 void StrCpy4(char *dest, const char *src) {
72   // dest++: dest, dest = dest + 1
73   // dest[0]
74   // *dest++: *dest, not *(dest + 1)
75   while ((*dest++ = *src++) != '\0');
76 }
77
78 // NOT recommended!
79 void StrCpy5(char *dest, const char *src) {
80   // '\0': null character, 0
81   while ((*dest++ = *src++));
82 }
83
84 char *StrCpyStd(char *dest, const char *src) {
85   for (char *s = dest; (*s++ = *src++) != '\0';);
86   return dest;
87 }
```