Homework

27

C-String Functions

Write the following two C-string functions "from scratch". Do not use any functions from the standard library, and you cannot use the C++ string class. Read through, and follow the instructions below.

1. The *reverse* Function

This function reverses (in place) a C-style character string `s`. You must use pointers and the increment and decrement operators only.

- You may not use any library functions.
- You may not allocate any other array storage
- You may not use pointer arithmetic (that is `p + n`)
- You may not use array notation in your function.

In other words, there will be no `int` variables or literals in your function: only pointers along with assignment, comparison, pointer subtraction, `++` and/or `--`; you may use:

- a temporary `char` variable to exchange the elements of the string
- a temporary pointer.

2. The *findStr* Function

Implement your version of the standard C-library function `strstr()`. So that it doesn't conflict with the names in the standard library, use `findStr()`.

```
1 | const char * findStr(const char *s1, const char *s2);
```

The findStr prototype.

The function:

- finds the **first occurrence** of the entire string **`s2`** inside the string **`s1`**
- returns a pointer to the first occurrence.
- If no match is found, then return the C++ **`nullptr`**.
- If **`s2`** points to a string of zero length, then return **`s1`**.

Here are some examples:

```
1  findStr("cowtown", "ow");  // returns &s1[1]
2  findStr("cowtown", "own"); // returns &s1[4]
3  findStr("cowtown", "two"); // returns nullptr
```

Using the findStr function.

You may use pointer notation or array notation. However, **do not use any functions from the standard library.** That includes **`strlen()`** or the C++ **`string`** class.

Be sure to **`make submit`** to turn in your code for credit **before the deadline**. As always, if you run into problems, bring your questions to the Discussion Board, or come to my office hour.