7. PROGRAMMING ASSIGNMENT 7:
Read: Chapter 10
Programming: Name your program p7.c
DUE: Wednesday September 6, 2017 @ 6am
This program will use character strings, functions, and pointer notation to reverse the input string in place (change original sentence).

a) Each function should have a short header comment describing its purpose.

b) `main()` reads input and prompts for program repetition.
   `fgets()` will read in an entire line of input where "MAX_STR" is a symbolic constant value of 81. e.g. "fgets(inputStr, MAX_STR, stdin);"
   Note: `fgets()` includes the trailing "newline" character. Refer to your Character Strings handout. Note: "getchar();" is not needed after `fgets()`.
   `main()` calls `strRev()` and passed one character string with this declaration:
   ```c
   void strRev(char *pS);
   ```
   ```c
   char inputStr[MAX_STR];              // Holds input string
   char s[MAX_STR];                     // Holds input stripped of newline
   ...
   printf ("Enter a sentence: ");
   fgets(inputStr, MAX_STR, stdin);   // Read input line including newline
   len = strlen(inputStr);            // Length of input string incl. newline
   strncpy(s, inputStr, len-1);       // Stripped ending newline char
   s[len-1] = '\0';                   // Delimit with null char
   ```

c) In `strRev()`, use pointer notation within ALL code of function definition.
   A local temporary character string, `tmpStrg`, can hold the input sentence. e.g. "Proud to be UCSD" --> "DSCU eb ot duorP"
   Remember: A character string is terminated with the end of string sentinel (null character '\0'). The original sentence IS modified.

d) Your program must loop so the user can continue to generate and view the output until the user responds with 'n' or 'N' to the offer "Want to reverse another sentence?". NOTE: Your program will end ONLY with the input of 'n' or 'N'.

HINTS: Solve this problem in small steps. Compile and test gradually.
1) In `main()`, write the code for steps a, b), d) and f) above. Test.
2) In `strRev()`, copy (strcpy) the input sentence "inputStr" into `tmpStrg`.
   Assign from `tmpStrg` beginning element into ending element of original word. You may want to include `<string.h>` for function `strcpy()`, `strncpy()`, `strlen()`.
   Use `printfs` to help debug temporary strings.
3) Add a "getchar()" in `main()`:
   ```c
   printf ("Want to reverse another sentence? ");
   answer = getchar();
   ```
   ```c
   getchar();
   ```
4) Complete error checking. Test. Test. Test. DONE!

PA#7 SAMPLE OUTPUT:
```c
Enter a sentence: Summer Fun Time!
"Summer Fun Time!" reversed is "!TunF miTeS"
Want to reverse another sentence? X

Enter a sentence: Yaaayyy! Last Program=)
"Yaaayyy! Last Program=" reversed is ")=Porsgta !Yaaayyy"
Want to reverse another sentence? Z

Enter a sentence: Angus Arnold Seth
"Angus Arnold Seth" reversed is "Ahts dlonrA ghnsA"
Want to reverse another sentence? N
```

SAMPLE INPUT (is in bold type above)
Submit the final version of your program as “p7.c”.

Verify you saved your work in the Documents - cs5v HOME directory.