Programming 1 – Exercise Set 4

1. Write a function that measures length of a null-terminated character string. Return a number of characters without the trailing `'\0'`.
   ```
   int  mystrlen (const char *src );
   ```
2. Write a function that will copy one null-terminated string to another.
   ```
   void mystrcpy ( char *dest, const char *src);
   ```
3. Write a function that compares one null-terminated string with another. Function returns 0 if strings are equal, or -1 otherwise.
   ```
   int mystrcmp ( char *dest, const char *src );
   ```
4. Write a function that compares at most n first elements from one null-terminated string with anothe. Function returns 0 if strings are equal, or -1 otherwise. Remember that the strings might be shorter then n characters.
   ```
   int mystrncmp ( char *dest, const char *src , size_t n);
   ```
5. Write a function that concatenate one null-terminated string to another.
   ```
   char* mystrcat ( char *dest,  const char *src );
   ```
6. Write a function that returns address of the first occurrence of a given character in a null-terminated string. Return a NULL pointer if the character is not in the string.
   ```
   char* mystrchr (const char *src, int  c);
   ```
7. Write a function which checks if a null-terminated character string is a palindrome, i.e. the sequence of characters in the string read in reverse order give the same string. Return a non-zero value if the string is a palindrome, or 0 otherwise.
8. Write a function which removes a character from a null-terminated string at a given position.
   ```
   void strrmidx(char *src, int i);
   ```
9. Write a function which removes first occurrence of a character in a null-terminated string.
   ```
   void strrmchr(char *src, char ch);
   ```
10. Modify the previous function to remove every occurrence of a given character.
11. Write a function which removes from a null-terminated string every uppercase letter.
12. Write a function which removes from a null-terminated string a substring indicated by starting position and length.
    ```
    void srtrmsub(char *src, int start, int length);
    ```
13. Write a function which removes from a string every occurrence of another string.
    ```
    void strrmstr(char *src, const char *toremove);
    ```
14. Write a function which replaces in a null-terminated string every occurrence of a given character with another. Return the number of characters that were replaced.
    ```
    int strrplcch(char *src, char toreplace, char replacewith);
    ```
15. Write a function which changes all lowercase characters in string to uppercase.
16. Write a function which replaces in a null-terminated string a substring indicated by start position and length with another string.
    ```
    void strrplcsub(char *src, int start, int length, const char replacement);
    ```
17. Write a function which replaces in a null-terminated string every occurrence of a given string with another. Return the number of replacements.
    ```
    int strrplcstr(char *src, const char *toreplace, const char *replacewith);
    ```
18. Write a function which replaces all numbers (sequences of digits) less then or equal to 12 with their written representation, e.g. “44 3 123 12” into “44 three 123 twelve”.

Programming 1 – Exercise Set 4 (B Side)

1. Write a function with two parameters: an array of double values A and an integer n, which is the length of the array. Function should sort the values in the array using the selection sort method.

2. Write a function with two parameters: an array of double values A and an integer n, which is the length of the array. Function should sort the values in the array using the insertion sort method.

3. Write a function with two parameters: an array of double values A and an integer n, which is the length of the array. Function should sort the values in the array using the bubble sort method.