Assignment : 9-14

Notes:

a) You are not allowed to use functions of **string.h** header file.

b) If you are intelligent enough, you can avoid repetitive code and write a group of function in a single program.

c) Try to write efficient code (less execution time and/or less space)

d) Due date: Just After I mid term. CR will be responsible to collect the hard copies and submit to me with the list who have submitted (no soft copies will be allowed).

1. Write following function for the string.
   a) length of a string
      - int xstrlen(char * str)
   b) Appends the string pointed by *src* to the end of the string pointed to by *dest*.
      - char * xstrcat(char *dest, const char *src)
   c) Find first occurrence of character c in string
      - char * xstrchr(const char *str, int c)
   d) Find last occurrence of character c in string.
      - char *xstrrchr(const char *string, int c)
   e) Compares the string pointed by *str1* to the string pointed to by *str2*
      - int xstrcmp(const char *str1, const char *str2)
   f) Copies the string pointed by *src* to *dest*.
      - char *xstrcpy(char *dest, const char *src)
   g) Copies up to n characters from the string pointed by *src* to *dest*.
      - char *xstrncpy(char *dest, const char *src, size_t n)
   h) locates the first occurrence of the string s2 in string s1.
      - char *strstr(const char *s1, const char *s2)
2. WAP which reads a string and reverse it.
3. WAP which reads a string and determines the number of alphabets, digits and special characters in that string.
4. WAP which reads a string and determines the number of vowels.
5. WAP which reads a string and convert it into uppercase.
6. WAP which reads a string and convert it into lowercase.
7. WAP to check whether the given string is palindrome or not.
8. WAP which reads a string and count no. of words.
9. WAP which reads a string and reverses each word of the string.
10. WAP which reads a string and convert the string as following.
11. Character conversions and testing: ctype.h: ctype.h header file contains many useful functions to convert and test *single* characters. The common functions prototypes are as follows:

Character testing:

int isalnum(int c) – return True if c is alphanumeric.

int isalpha(int c) -- return True if c is a letter.

int iscntrl(int c) -- return True if c is a control character.
int isdigit(int c) -- return True if c is a decimal digit
int isgraph(int c) -- return True if c is a graphical character.
int islower(int c) -- return True if c is a lowercase letter
int isprint(int c) -- return True if c is a printable character
int ispunct (int c) -- return True if c is a punctuation character.
int isspace(int c) -- return True if c is a space character.
int isupper(int c) -- return True if c is an uppercase letter.
int isxdigit(int c) -- return True if c is a hexadecimal digit

Write your own version of functions for all above ptototypes.