# String exercises

1. Write a function that checks the number of repetitions of the abc(char* sub) character sequence in the stringchar* str, returns the number of repetitions ofsub.

```
int countSubString(char* str, char* sub);
```

2. Write a program that receives 3 strings from the user, chains all three of them into one string and displays it on the screen. Assume that each of the three received strings is no more than 15 in size.

```
char* addStringsTogether(char* str1, char* str2,  char* str3);
```

3. String inversion: Write a function that takes a string,creates a new string that contains the first string from the end to the beginning. That is, if the user typed "Shalom" the second string would be "molahS". Assume that strings are no more than 15 years long.

```
char* flipStr(char* str);
```

4. Write a function that takes a sentence and returns the sentence with each word inverted in its letter order.
   Example:
   input : This is an example
   return : sihT sin a elpmaxe

```
char* flipWords(char* str);
```

5. Delete spaces from an input string - Write a function that takes a string, and deletes all spaces from it. Assume that strings are no longer than 15.

```
char* removeBlanks(char* str);
```

6. Write a function named strlen_without_c that takes a string, and prints the number (amount) of characters in it other than the letter 'c'.

```
int strlen_without_c(char* str, char c);
```

7. Write a function where you use a function namedstrcpy_without_cthat copies one string to another, except for any instance of the letter 'c'.

```
char* strcpy_without_c(char* str, char c);
```

8. Write a functionthat takes a string and returns 1 if the word is a palindrome, 0 if the word is not a palindrome.

```
int isPaly(char* str);
```

9. Write a function that accepts a sentence and turns the first letter of each word in the sentence into a capital letter, if the letter is already large there is no need to update it.

```
char * efriFirst (char * str);
```

10. Write a program that accepts a string and returns the highest number of times one letter in the string appears.

```
int mostCommonLetter(char* str);
```

11. Write a function that takes a string and returns the number of words in a string that have more uppercase letters than lowercase letters.

```
int moreUpperThanLower(char* str);
```