Consider the header file "functions.h":

```
char* str_replace(char* search, char* replace, char* orig);
char** explode(char* delimiter, char* orig, int* count);
char* implode(char* glue, char** pieces, int count);
```

1. Exercise 1:

1. Launch the terminal
2. Create a new directory with the name "Lab09" inside "CSC215"
3. Write a C file "functions.c" that contains all functions defined in this lab
4. Write the function str_replacewhich takes a string `orig`, search for all occurrences of the substring `search` and replaces each of them with the string `replace`.
5. Plan:
   - find out the number of occurrences of `search` in `orig`
   - prepare a buffer big enough to hold the resulting string
   - copy from `orig` to the new buffer until the next occurrence of search
   - copy the string `replace` to the new buffer
   - repeat until there is no more occurrences of `search`

2. Exercise 2:

1. Write the function explode which returns an array of strings, each of which is a substring of `string` formed by splitting it on boundaries formed by the string `delimiter`.
2. Plan:
   - find out the number of substring by counting the number of occurrences of the delimiter
   - prepare an array or pointers of the appropriate size
   - start tokenizing the string and add the tokens to the array

3. Exercise 3:

1. Write the function implodewhich returns a string representation of all the array `pieces` elements in the same order, with the `glue` string between each two consecutive elements.
2. Plan:
   - prepare an array or pointers of the appropriate size
   - copy the next element from the array to the buffer
   - copy the glue string from the array to the buffer
   - repeat for all array elements

Tips:

- Check the man pages for: `strlen`, `strcpy`, `strncpy`, `strstr`, `strtok`, `strdup`
- You can use the program `"test.c"` to test your functions.

```
#include <stdio.h>
#include "functions.h"
int main(){
  int count, i;
  char* str = "The white cat invited the red cat to a caterpillar lunch";
  printf("%s\n\n", str_replace("cat", "tiger", str));
  char** carr = explode(" ", str, &count);
  for (i=0; i < count; i++) printf("%s\n", carr[i]);
  printf("\n%s\n", implode("|", carr, count));
  return 0;
}
```