

FEEG6002 Advanced Computational Methods 1:

Laboratory-Assignment 4

Contents

- 1 Training: The length of a string
- 2 Laboratory: Reversing a string (=char array)

Prerequisites: Strings

This lab session should allow you to practice many of the skills taught so far with specific focus on getting to grips with manipulating strings.

1 Training: The length of a string

- In a file with name training4.c, write a C function long string_length(char s[]) which returns the length of a string s as integer type long.

For example, the string "Hello World" has a length of 11, the string "x" has a length 1 and the string "line 1\tline 2\n" has the length 14.

You are invited to use the template training4.c to write and test your function.

training4.c

```
#include <stdio.h>
```

```
/* function prototype */  
long string_length(char s[]);
```

```
int main(void) {  
    char s1[]="Hello";  
    char s2[]="x";  
    char s3[]="line 1\tline 2\n";  
  
    printf("%20s | %s\n", "string_length(s)", "s");  
    printf("%20ld | %s\n", string_length(s1), s1);  
    printf("%20ld | %s\n", string_length(s2), s2);  
    printf("%20ld | %s\n", string_length(s3), s3);  
    return 0;  
}
```

As usual, submit your work by email to feeg6002@soton.ac.uk with subject line training 4 to receive feedback. The tests will only test the function string_length. You can thus modify the main function as you like to support your code writing, debugging and testing. However, as usual, your file should compile without errors and warnings when using the recommended switches; this includes the main function.

2 Laboratory: Reversing a string (=char array)

- In a file with name lab4.c, write a C function void reverse(char source[], char target[]) that copies all the characters in the character array source into the character array target but in reverse order, i.e. the first character in source should be the last character in target.

You are invited to use the template lab4.c to write and test your function.

lab4.c

```
#include <stdio.h>
#define MAXLINE 1000 /* maximum length of string */

/* function prototype */
void reverse(char source[], char target[]);

int main(void) {
    char original[] = "This is a test: can you print me in reverse character order?";
    char reversed[MAXLINE];

    printf("%s\n", original);
    reverse(original, reversed);
    printf("%s\n", reversed);
    return 0;
}

/* reverse the order of characters in 'source', write to 'target'.
Assume 'target' is big enough. */
void reverse(char source[], char target[]) {
```

If you want to call the function string_length from your code, you need to copy it into the lab4.c file. There are other solutions that don't require using string_length - we don't give preference to either approach.

As usual, submit your work by email to feeg6002@soton.ac.uk with subject line lab 4 to receive feedback. The tests will only test the function reverse. You can thus modify the main function as you like to support your code writing, debugging and testing. However, as usual, your file should compile without errors and warnings when using the recommended switches; this includes the main function.