

Programming 1 – Exercise Set 8

1. Write a function that with a single parameter – a name of the output file. Function should read lines of text from the standard input and store them in the file until an empty line is read. Function should return the number of lines written. How would you modify the function so that the output file is not erased each time the function is called?
2. Write a function with a single parameter – a name of an input file. Function should read characters from the file one by one and print them to the standard output. However, if a character is not a whitespace, letter or digit, an underscore should be printed instead.
3. Define a structure `Student` with three members: a first name as an array of 10 characters, a last name as an array of 20 characters and semester as an integer. Write a function which takes three parameters: a file name, an array of `Student` variables and the size of that array. The function should write to the output file the size of the array in the first line, and after that in separate lines the elements of the array in the format of:

```
first_name last_name semester
```

If the output file doesn't exist it should be created, otherwise it's content should be discarded.
4. Write a function which takes a file name as a parameter. The function should expect a file written by the call to the function described in the previous exercise. First it should read the number of students, then allocate a new array of the appropriate size and read the values of the elements from separate lines of the file. That array should be returned by the function. Make sure that the input file exist and if it doesn't print out an error message and return a NULL pointer. Write a fragment of code demonstrating use of functions from previous and this exercise.
5. Write two functions similar to those described in previous two exercises, but instead of writing the values as formatted text in separate lines, the size of the array and its elements should be written and read as raw bytes. Consider also reading from selected position from file.
6. Write a function with a single parameter – a name of a file. Function should read characters from the file one by one. If a character is not a whitespace, letter or digit, function should overwrite it in the file with an underscore.
7. Write a program printing on the screen name of the program and all other arguments passed using command line.
8. Write a program expecting two command line arguments (e.g. *prog_name 234 x*) – an unsigned decimal integer and a format (*x*, *d* or *o*). Program should print the integer using the specified format (hexadecimal, decimal or octal respectively).
9. In main declare a 3x5 matrix (as a two-dimensional array) of double values. Write a function that will calculate the average value in each row and return the largest average.
10. Modify the function from task 9 to accept a matrix of arbitrary dimensions.
11. Modify the code from task 10 so that the matrix is an array of pointers to rows which are allocated dynamically.