Riccardo Cappuzzo
S191436

System Programming

LAB 02

Exercise 1

The source code is in file ex1.c
The sieve algorithm can be improved in order to have more efficiency and less computations.

Exercise 2

The source code is in file ex2.c

Exercise 3

The source code is in file ex3.c

Exercise 4

The source code is in file ex4.c
The solution proposed has the parent process waiting for the return value of the child process.

Exercise 5

The source code is in file ex5.c
I used the execl functions from the exec family since I knew in advance what were the commands I was supposed to use, so I didn't have to create an array of strings.

Exercise 6

The source code is in file ex6.c

Exercise 7

The source code is in file ex7.c
For some reason, if in the reading process I use the whole buffer of 512 bytes I get weird behaviors, so I needed to use 1 character less. I suppose it is a problem caused by the terminator character.

Exercise 8

The source code is in file ex8.c

Exercise 9

The source code is in file ex9.c
The average operation is performed by the average program. The source for average is in the file “average.c” . I am generating a file with up to MAX_RAND random numbers whose size ranges from 0 to MAX_VALUE-1. I am using the execl functions here because, like in exercise 5, I already know what commands I need to use and with which parameters.

Exercise 10

The source code is in file ex10.c
I need to read the buffer two times in order to find the number of commands I need to store and the commands themselves. I use the strtok() function and so I need an additional buffer because the

starting one is broken when using strtok.

Exercise 11

The file ex11.c uses a pipe in order to perform interprocess communication. A child process is generated, which computes the power of 2 needed and then writes the result on the pipe. Once this is done, the child process exits and the parent (which is waiting for the result of the child) reads from the pipe the value. The value is then added to the sum and the sum is shown on screen. Error control is performed when the pipe is created and on the input number.
I've also put another version of the script (ex11bis.c). It is a wrong solution, it employs the exit function as a “fake return” to the parent function: the result of the power is used as a return value and then added to the sum in the parent. This only works for return values smaller than 2^8.

Exercise 12

The source code is in file ex12.c
I use pipes in order to pass the value computed by the child processes to the parent process. The result is then assembled in the parent, which doesn't need a wait function since it reads from the pipe and so it is stopped by it as long as something is writing (i.e. children are still computing).