Task 1: atoi(), atof() functions- Run and understand following example.

<https://www.tutorialspoint.com/c_standard_library/c_function_atoi.htm>

<https://www.tutorialspoint.com/c_standard_library/c_function_atof.htm>

Parameterizing the main function

int main(argc, char argv \*[])

-argc (argument counter): contains the number of arguments passed on to the program plus one; this means that a program run without any arguments will have an argc parameter value equal to 1

-argv (argument values): an array of pointers to strings containing the arguments supplied to the program; they’re stored in the following way:

argv[0] contains the name of the running program

argv[1] contains the string passed to the program as the first argument

argv[n] contains the string passed to the program as the n-th argument

Task 2: Run and understand following example.

#include <stdio.h>

main(int argc, char \*argv[]) {

printf("%d\n", argc);

return 0;

}

Run program the following way.

root@debian:/home/gerelee/lab11# ./a.out

1

root@debian:/home/gerelee/lab11# ./a.out imp programming

3

#include <stdio.h>

int main(int argc, char \*argv[]) {

int i;

printf("%d\n", argc);

printf("What is in argv parameter? \n");

for(i=0; i<argc; i++)

{

printf("argv[%d] %s \n",i,argv[i]);

}

return 0;

}

Run program the following way.

root@debian:/home/gerelee/lab11# ./a.out 1 2

3

What is in argv parameter?

argv[0] ./a.out

argv[1] 1

argv[2] 2

#include <stdio.h>

#include <string.h>

int main(int argc, char \*argv[]) {

int i;

if(argc == 1) {

printf("usage: args arg1 arg2 arg3 ...\n");

return 1;

}

if(argc == 2 && strcmp(argv[1],"-v") == 0) {

printf("args version 1.0, C language course, 2012\n");

return 2;

}

for(i = 0; i < argc; i++)

printf("%s\n", argv[i]);

return 0;

}

Runing example with –v parameter:

root@debian:/home/gerelee/lab11# ./a.out -v

args version 1.0, C language course, 2012

Task 3 : Main function parameters

Familiarize the student with:

* Main function parameters
* Simple arithmetic
* Printing on screen

Scenario

Write a program that allows the user to pass the parameters to be executed and compute the results of some mathematical operations. Your program should support the following operations:

add

sub

mul

All operations require an additional two arguments. Some examples of program calls inlude:

./a.out add 1 3

./a.out sub 2 3

./a.out mul 2 5

When there are no parameters, the parameters contain the wrong numbers or a parameter is invalid, the program should print the message: "Wrong parameters" (or you can think about your own message with regard to the proper form of program execution). The first parameter must be one of the three previous strings, while the second and third parameters must be integer numbers. To find out which of the commands has been passed, you may use the strcmp function. Your version of the program must print the same result as the expected output. This is one of the tasks which can only be executed in your personal environment(CLI) to provide parameters. If you use geany, please use terminal.

Example input

add 1 3

Example output

add 1 3: 4

Example input

sub 2 3

Example output

sub 2 3: -1

Example input

mul 2 5

Example output

mul 2 5: 10