In the first program, the M. A. I. N. function has two arguments, called A. R. G. C. and A. R. G. Vee.

This lecture explains what they are.

Let us consider this example.

This program creates an integer as the index for an array. An array is an ordered list of things. Each element in an array has an index.

The next line prints A. R. G. C's value.

In print F., percentage D. means the value of an integer.

A R. G. C. means the number of arguments given to the program. The value is at least one.

The first argument is the program's name. Since every program needs a name, A. R. G. C. must be one or larger.

In a C program, FOR means getting the values within a range. In this example, the index's value starts from zero, then it becomes one, then two, until it is A. R. G. C. minus one.

Please notice that the second part of FOR uses less than. This means that the value of I.N.D. has to be smaller than A. R. G. C.

.

For each value of the index, the corresponding argument's value is printed. A. R. G. Vee. is an array of values. Each element is a string. In C programs, a string is like a word, a phrase, or a sentence.

This line prints two values. The first is an integer and it is the value of the index.

The second is a string.

Please remember that the index starts from zero, not one. The largest value of the index is A. R. G. C. minus one. It cannot be A. R. G. C.

.

Now, let's run the program and understand how it works.

The first line prints "A R G C equals to one". This means the program has one input argument.

The second line prints the only argument. This is the first element of the A R G Vee. The element is the name of the program.

Next, let's add some more arguments.

We can enter E C E. two six four. If letters are connected without space, they are treated as a single word. Space separates two words. We can use one or multiple spaces to separate words.

The program’s output says there are three arguments. The first is the program’s name. The second is the word E C E. . The third argument is two six four as a single word.

The next example has even more arguments.

You can add more spaces between the arguments. One space is as good as many. More spaces make no difference.

You may be curious who calls the main function and how the arguments are given to the main function. In Linux, it is the run time system that calls the main function. More specifically, the bash calls the program.

The next lecture explains how to use A. R. G. C. and A. R. G. Vee. to provide inputs to a program.