

How to Count Variable Numbers of Arguments in C?

C supports variable numbers of arguments. But there is no language provided way for finding out total number of arguments passed. User has to handle this in one of the following ways:

- 1) By passing first argument as count of arguments.
- 2) By passing last argument as NULL (or 0).
- 3) Using some printf (or scanf) like mechanism where first argument has placeholders for rest of the arguments.

Following is an example that uses first argument *arg_count* to hold count of other arguments.

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

// this function returns minimum of integer numbers passed. First
// argument is count of numbers.
int min(int arg_count, ...)
{
    int i;
    int min, a;

    // va_list is a type to hold information about variable arguments
    va_list ap;

    // va_start must be called before accessing variable argument list
    va_start(ap, arg_count);

    // Now arguments can be accessed one by one using va_arg macro
    // Initialize min as first argument in list
    min = va_arg(ap, int);


    // traverse rest of the arguments to find out minimum
    for(i = 2; i <= arg_count; i++) {
        if((a = va_arg(ap, int)) < min)
            min = a;
    }

    //va_end should be executed before the function returns whenever
    // va_start has been previously used in that function
    va_end(ap);

    return min;
}
```

```
int main()
{
    int count = 5;

    // Find minimum of 5 numbers: (12, 67, 6, 7, 100)
    printf("Minimum value is %d", min(count, 12, 67, 6, 7, 100));
    getchar();
    return 0;
}
```



Minimum value is 6