

Download Ebook Fisika Dasar I. fbc29784dd. The Known Universe And Its Origins: Astronomy, Cosmology And Fundamental Physics - Tipler, Fisika Untuk Sains dan Teknik (terjemah oleh.Q: How to make a #define of function calls which have trailing arguments? I have functions with trailing arguments like so: void myFunction(const unsigned char \*src, int src\_len, unsigned char \*dst, int dst\_len); When I try to use the function like this: void myFunction(const unsigned char \*src, int src\_len, unsigned char \*dst, int dst\_len, int test); I get this error: "undefined reference to `myFunction'" Which is not surprising - but I don't know how to make a #define of this. A: I'm assuming your code is quite simple, and that if you're using gcc, you're also using C++: #ifdef \_\_cplusplus #define myFunction(...) myFunction(\_\_VA\_ARGS\_\_, \_\_VA\_ARGS\_\_) #else #define myFunction(...) myFunction(\_\_VA\_ARGS\_\_, \_\_VA\_ARGS\_\_) #endif (I also assume that the explicit arguments are just dummy arguments for the function.) In C++, you can declare a function as taking an arbitrary number of arguments: void myFunction(const unsigned char \*src, int src\_len, unsigned char \*dst, int dst\_len, int test); Using the above is just a shorthand: you write myFunction(...); instead of myFunction(.....); and you can write void myFunction(const unsigned char \*src, int src\_len, unsigned char \*dst, int dst\_len, int test); (You don't have to use the comma though.) Because \_\_VA\_ARGS\_\_ expands to all the remaining arguments, you can put

[Download](#)



