

[Learning](#)[Chat](#)[Gitea](#)

Name - Login



My Printf

[Subject](#)[2 Solutions](#)[Additional
Resources \(0\)](#)

My Printf

Technical details	
Submit file	my_printf.c

NAME

```
int my_printf(char * restrict format, ...);
```

Prolog

The very first thing we do when learning a programming language is to write "Hello World !" on the screen. Well guess what ? This custom comes from a notebook called ...

Programming in C: A Tutorial !

Unfortunately for you, this project will bring you back in the 1970s, before the existence of this book, and you will have to write your own function to greet the world.

Control Center

[Group formation](#)[In Progress](#)[Submitted](#)[Peer review](#)

Finished: approved

[Go To DoCode](#)

Access:

[READ](#)[WRITE](#)[Go To Gitea](#)[Keep Working On This Solution](#)

**Also working on
the project**

Fortunately for you, you will strengthen your programming skills and learn new programming concepts: variadic functions and variable argument lists. Recoding printf is also a good exercise to learn to have a well structured code organisation. Modularity is the key here since printf has a lot of different options that you will have to recode yourself. Adding a new option feature should be easy and smooth if the code is well designed !

DESCRIPTION

The **my_printf()** function produce output according to a format as described below. The *my_printf()* functions write output to stdout, the standard output stream;

This function writes the output under the control of a format string that specifies how subsequent arguments (or arguments accessed via the variable-length argument facilities of stdarg(3)) are converted for output.

The format string is composed of zero or more directives: ordinary characters (not %), which are copied unchanged to the output stream; and conversion specifications, each of which results in fetching zero or more subsequent arguments.

Each conversion specification is introduced by the % character. The arguments must correspond properly (after type promotion) with the conversion specifier. After the %, the following appear in sequence:

- doux** The *int* (or appropriate variant) argument is converted to signed decimal (**d**). unsigned octal (**o**), unsigned decimal (**u**), unsigned hexadecimal (**x**).
- c** The *int* argument is converted to an unsigned char, and the resulting character is written.
- s** The *char ** argument is expected to be a pointer to an array of character type (pointer to a string). Characters from the array are written up to (but not including) a terminating NUL character.
- p** The *void ** pointer argument is printed in hexadecimal.

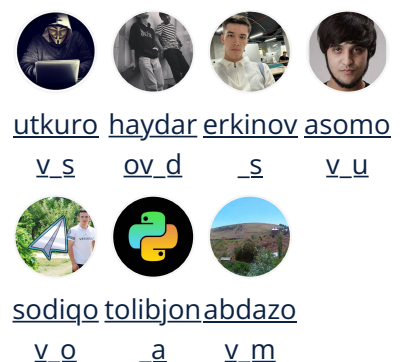
Requirements

Your code must be compiled with the flags **-Wall -Wextra -Werror**.

the project



Just finished



Type
Project

Group
Size

1
Participant

Hint(s)

- man printf
- man 3 stdarg
- Watch out for **memory leaks** !
- You can test your code against memory errors by compiling with the debugging flags **-g3 -fsanitize=address**
- **Global variables are strictly FORBIDDEN**
- **Static variables are strictly FORBIDDEN**

Authorized function(s)

- write(2)
- malloc
- free
- va_start, va_arg, va_copy, va_end (see man 3 stdarg)

(Obvious) Unauthorized functions

printf and co. (fprintf, sprintf, snprintf, asprintf, dprintf, vprintf, vfprintf, vsprintf, vsnprintf, vasprintf, vdprintf)

Epilog

Don't forget you can ask questions on the chat !

Good luck!

Review
system

[Peer Review](#)

Difficult
y

[Initiation](#)

Average
e
duration
n

1
Week

Project's Metadata

Project

id: 15

name: [my_printf](#)

visible: [True](#)