



## ft\_printf

Porque ft\_putnbr() e ft\_putstr() não são suficientes

*Resumo: O*

*objetivo deste projeto é bastante simples. Você recodificará printf().*

*Você aprenderá principalmente como usar um número variável de argumentos. Quão legal é isso??*

*Na verdade, é muito legal :)*

*Versão: 10*

# Conteúdo

<b>EU</b>	<b>Introdução</b>	<b>2</b>
<b>II</b>	<b>Instruções Comuns</b>	<b>3</b>
<b>III</b>	<b>Parte obrigatória</b>	<b>5</b>
<b>IV</b>	<b>parte bônus</b>	<b>7</b>
<b>V</b>	<b>Submissão e avaliação por pares</b>	<b>8</b>

# Capítulo I

## Introdução

Você descobrirá uma função C popular e versátil: `printf()`. Este exercício é uma ótima oportunidade para melhorar suas habilidades de programação. É de dificuldade moderada.

Você descobrirá **funções variadas** em C.

A chave para um `ft_printf` bem-sucedido é um código bem estruturado e extensível.



Depois que esta tarefa for aprovada, você poderá adicionar `ft_printf()` à sua `libft` para poder usá-la em seus projetos C escolares.

## Capítulo II

### Instruções Comuns

- Seu projeto deve ser escrito em C.
- Seu projeto deverá ser escrito de acordo com a Norma. Se você tiver arquivos/funções bônus, eles serão incluídos na verificação de norma e você receberá 0 se houver um erro de norma.
- Suas funções não devem encerrar inesperadamente (falha de segmentação, erro de barramento, liberação dupla, etc.) além de comportamentos indefinidos. Caso isso aconteça, seu projeto será considerado não funcional e receberá nota 0 na avaliação.
- Todo o espaço de memória alocado no heap deve ser liberado adequadamente quando necessário. Sem vazamentos será tolerado.
- Se o assunto exigir, você deve enviar um Makefile que irá compilar seus arquivos fonte para a saída necessária com as flags -Wall, -Wextra e -Werror, use cc, e seu Makefile não deve relinkar.
- Seu Makefile deve conter pelo menos as regras \$(NAME), all, clean, fclean e ré.
- Para entregar bônus ao seu projeto, você deve incluir uma regra bônus em seu Makefile, que adicionará todos os diversos cabeçalhos, bibliotecas ou funções que são proibidas na parte principal do projeto. Os bônus devem estar em um arquivo diferente \_bonus.{c/h} se o assunto não especificar mais nada. A avaliação da parte obrigatória e da parte bônus é feita separadamente.
- Se o seu projeto permitir que você use sua libft, você deve copiar seus fontes e seu Makefile associado em uma pasta libft com seu Makefile associado. O Makefile do seu projeto deve compilar a biblioteca usando seu Makefile e depois compilar o projeto.
- Nós encorajamos você a criar programas de teste para o seu projeto, mesmo que este trabalho **não precise ser enviado e não receba notas**. Isso lhe dará a chance de testar facilmente seu trabalho e o de seus colegas. Você achará esses testes especialmente úteis durante sua defesa. Na verdade, durante a defesa, você é livre para usar seus testes e/ou os testes do colega que está avaliando.
- Envie seu trabalho para o repositório git designado. Somente o trabalho no repositório git será avaliado. Se Deepthought for designado para avaliar seu trabalho, isso será feito

após suas avaliações por pares. Se ocorrer um erro em qualquer seção do seu trabalho durante a avaliação do Deepthought, a avaliação será interrompida.

# Capítulo III

## Parte obrigatória

<b>Nome do programa</b>	libftprintf.a
<b>Entregar arquivos</b>	Makefile, *.h, /*.h, *.c, /*.c
<b>Funções</b>	NOME, tudo, limpo, fclean, re
<b>externas do Makefile.</b>	malloc, grátis, escrever, va_start, va_arg, va_copy, va_end
<b>Libft autorizado</b>	Sim
<b>Descrição</b>	Escreva uma biblioteca que contenha ft_printf(), um função que irá imitar o printf() original

Você precisa recodificar a função printf() da libc.

O protótipo de ft\_printf() é:

```
interno    ft_printf(const char *, ...);
```

Aqui estão os requisitos:

- Não implemente o gerenciamento de buffer do printf() original.
- Sua função precisa lidar com as seguintes conversões: cspdiuxX%
- Sua função será comparada com o printf() original.
- Você deve usar o comando ar para criar sua biblioteca.  
Usar o comando libtool é proibido.
- Seu libftprintf.a deve ser criado na raiz do seu repositório.

Você deve implementar as seguintes conversões:

- %c Imprime um único caractere.
- %s Imprime uma string (conforme definido pela convenção C comum).
- %p O argumento do ponteiro void \* deve ser impresso em formato hexadecimal.
- %d Imprime um número decimal (base 10).
- %i Imprime um número inteiro na base 10.
- %u Imprime um número decimal sem sinal (base 10).
- %x Imprime um número em formato hexadecimal (base 16) em minúsculas.
- %X Imprime um número em formato hexadecimal (base 16) maiúsculo.
- %% Imprime um sinal de porcentagem.

# Capítulo IV

## Parte bônus

Você não precisa fazer todos os bônus.

Lista de bônus:

- Gerencie qualquer combinação dos seguintes sinalizadores: '-0.' e a largura mínima do campo sob todas as conversões.
- Gerenciar todos os seguintes sinalizadores: '# +' (Sim, um deles é um espaço)



Se você planeja completar a parte bônus, pense na implementação de seus recursos extras desde o início. Dessa forma, você evitará as armadilhas de uma abordagem ingênua.



A parte bônus só será avaliada se a parte obrigatória for PERFEITA. Perfeito significa que a parte obrigatória foi feita integralmente e funciona sem mau funcionamento. Se você não passou em TODOS os requisitos obrigatórios, sua parte do bônus não será avaliada de forma alguma.



## Capítulo V

## Envio e avaliação por pares

Entregue sua tarefa em seu repositório Git normalmente. Somente os trabalhos dentro do seu repositório serão avaliados durante a defesa. Não hesite em verificar os nomes dos seus arquivos para garantir que estão corretos.

Depois que esta tarefa for aprovada, você poderá adicionar `ft_printf()` à sua `libft` para poder usá-la em seus projetos C escolares.

[illegible]