



## ft\_printf

ft\_putnbr()とft\_putstr()では不十分なため。

### 概要

このプロジェクトのゴールはとても簡単です。printf()を再コード化するので  
主に可変個数の引数を使うことについて学びます。なんてカッコイイんだろう？  
実際、かなりクールです:)

バージョン : 9.2

# 内容

I	はじめに	2
II	共通説明書	3
III	必須項目	5
IV	ボーナスパート	7
V	提出と相互評価	8

# 第一章 はじめに

よく使われる汎用性の高いC関数である`printf()`を発見することができます。この演習は、あなたのプログラミングスキルを向上させる絶好の機会です。難易度は中程度です。

C言語における**変項関数**を発見することができます。

`ft_printf`の成功の鍵は、よく構造化され、拡張可能なコードです。



この課題に合格したら、あなたの`libft`に`ft_printf()`を追加して、学校のCプロジェクトで使えるようにすることが許可されます。

## 第二章

### 共通事項

- プロジェクトはC言語で書かれている必要があります。
- あなたのプロジェクトはNormに則って書かれていなければなりません。ボーナスファイル/関数がある場合、それらはノームチェックに含まれ、内部にノームエラーがある場合は0が返されます。
- 未定義の動作とは別に、関数が予期せず終了する (セグメンテーションフォールト、バスエラー、ダブルフリーなど) ことがないようにしてください。このような場合、プロジェクトは非機能とみなされ、評価時に0点が与えられます。
- ヒープで確保されたメモリ空間は、必要なときに適切に解放されなければなりません。リークは許されません。
- 対象がそれを要求している場合、ソースファイルを要求された出力にコンパイルする Makefile を -Wall、-Wextra、-Werror フラグで提出し、cc を使用し、その Makefile は再リンクしてはいけません。
- Makefileには、少なくとも\$(NAME)、all、clean、fclean、および、(NAME)のルールが含まれていなければなりません。  
再
- プロジェクトにボーナスを回すには、Makefile にルールボーナスを含める必要があります、それによってプロジェクトのメイン部分で禁止されている様々なヘッダー、libraries または関数がすべて追加されます。ボーナスは、主体が何も指定しない場合は、別のファイル `_bonus.{c/h}` にする必要があります。必須部分とボーナス部分の評価は別々に行われます。
- プロジェクトで libft を使用できるようにする場合、そのソースと関連する Makefile を libft フォルダにコピーしておく必要があります。あなたのプロジェクトの Makefile は、その Makefile を使ってライブラリをコンパイルし、その後プロジェクトをコンパイルしなければなりません。
- テストプログラムは提出する必要がなく、採点もされませんが、プロジェクトのテストプログラムを作成することをお勧めします。テストプログラムを作成することで、自分の作品や仲間の作品を簡単にテストすることができます。このようなテストは、特にデフェンスの時に役に立ちます。実際、審査では、自分のテストや審査する仲間のテストを自由に使用することができます。
- 指定された git リポジトリに作品を提出する。git リポジトリにある作品だけが採点されます。Deepthought があなたの作品の採点を担当する場合、採点は以下のように行われます。



ft\_printfとft\_putnbr() と ft\_putstr() だけでは不十分なため。

---

相互評価後Deepthoughtの採点中に、あなたの作品のいずれかのセクションでエラーが発生した場合、評価は停止されます。

## 第三章 必須項目

プログラム名	libftprintf.a
ファイルを提出する	Makefile、*.h、*/*.h、*.c、*/*.c
メイクファイル	NAME、All、Clean、Fclean、Re
外部機能。	malloc、free、write。 va_start、va_arg、va_copy、va_end
リポート認可	はい
商品説明	ft_printf()を含むライブラリを書いてください。 という関数で、元の printf() を模倣します。

libcのprintf()関数を再コード化する必要があります。

ft\_printf()のプロトタイプです。

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

以下はその条件です。

- 本家printf()のバッファ管理は実装しない。
- この関数は次の変換を処理しなければなりません。
- 作成した関数は、オリジナルのprintf()と比較されます。
- ライブラリを作成するには、arコマンドを使用する必要があります。libtoolコマンドの使用は禁止されています。
- libftprintf.aは、リポジトリのルートに作成する必要があります。

ft\_printfとft\_putnbr() と ft\_putstr() だけでは不十分なため。

---

以下の変換を実施する必要があります。

- **c** 1文字を表示します。
- **s** 文字列を表示します (C言語の一般的な規約で定義されています)。
- **p** `void *` ポインタの引数は16進数で表示する必要があります。
- **d** 10進数 (10進数) を表示します。
- **i** 10進数の整数を表示します。
- **u** 符号なし10進数 (ベース10) を表示します。
- **x** 16進数 (ベース16) の小文字で数値を表示します。
- **X** 16進数 (ベース16) の大文字で数値を表示します。
- **%%** パーセント記号を表示します。



## 第四章 ボーナス パート

ボーナスを全部やる必要はない。

ボーナスリスト

- 以下のフラグの任意の組み合わせを管理する。'-0.'と、すべての変換の下のフィールドの最小幅を管理する。
- 以下のフラグをすべて管理する。'#+' (はい、そのうちの1つはスペースです)



ボーナスパートを完成させるつもりなら、最初からおまけ機能の実装を考えておくことです。そうすれば、甘い考えによる落とし穴を避けることができます。



ボーナスパーツは、必須パーツがPERFECTである場合にのみ査定されます。パーフェクトとは、必須パーツが統合的に行われ、誤動作することなく動作することを意味します。必須項目をすべてクリアしていない場合、ボーナスパーツの評価は一切行われません。

## 第五章

# 提出と相互評価

通常通り、Git リポジトリに課題を提出してください。リポジトリ内の作業だけが、デ  
ィフェンスで評価されます。ファイル名が正しいかどうか、遠慮なく再確認してく  
ださい。

この割り当てに合格したら、`ft_printf()`をあなたの  
`libft`は、学校のC言語プロジェクトで使えるように。