

REP03

所属：京都コンピュータ学院情報処理科

作成者：ラーション・ステファン、学生番号：E20C4033

Git：<https://github.com/Stefanperlarsson/REP03>

概要

- [表紙](#)
 - [概要](#)
 - [外部仕様](#)
 - [内部設計・プログラムの詳細](#)
 - [流れ](#)
 - [特徴](#)
 - [関数の説明](#)
 - [テスト方法](#)
 - [まとめ](#)
-

外部仕様

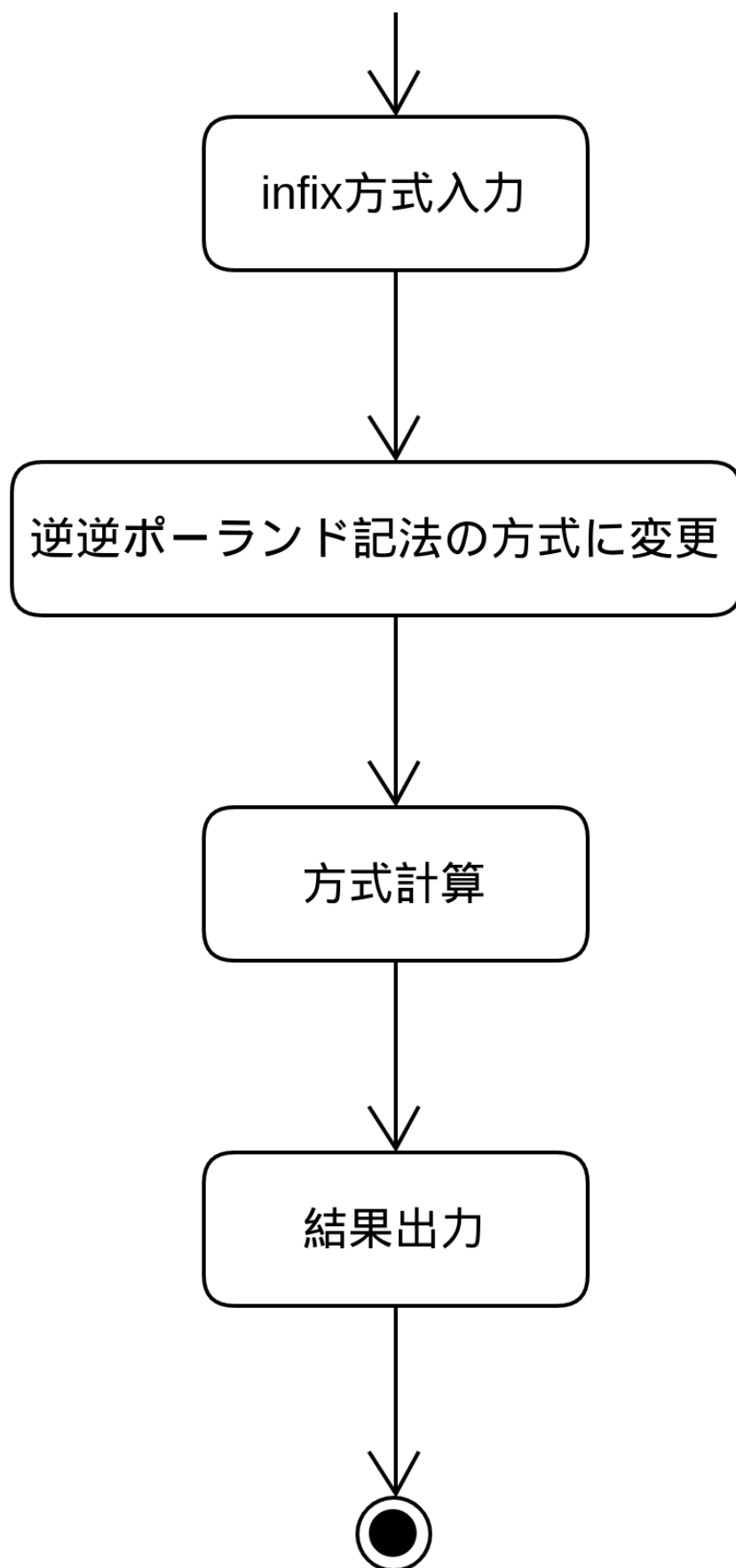
C言語でプログラミングしたInfix計算式を採算できるソフトです。

端末を使用して、infix系の方式を記入すると、postfix（逆ポーランド記法）に変更して、計算します。

内部設計・プログラムの詳細

流れ





特徴 - 必修なところ以外では、自分で頑張ってテッポアップしました。

- InfixからPostfix変更できる
- 自分のstackコード (struct使用、複数stack可能)

関数の説明

ファイル名	関数名	引数	戻り値	説明
REP03.c				
	main	(なし)	(なし)	メイン
stack.c				
	init	(なし)	stack *	stack作成
	push	stack *, char	(なし)	stackに追加
	pop	stack *	char	stackから読み取る
infix.c				
	convert	char *	(なし)	InfixからPostfixに変更

テスト方法

コンパイル

```
gcc ./REP03.c ./stack.c ./infix.c -o REP03
```

実行

```
./REP03;
```

使い方 (例)

```
C 言語実習 2 REP03 E20C4033 ラーション・ステファン
Infixを入力 ( 例 : 「 ( 1 - ( 2 + 3 ) ) 」 ) : ( 1 - ( 2 + 3 ) )
Infix: ( 1 - ( 2 + 3 ) )
Postfix(逆ポーランド記法): 123+-
答え: -4
```

まとめ

印象

今回も色々な勉強になりました。得にstackの実現は役に立てると思います。

改良の余地

- Infixの入力確認
- 一行以上数可能
- 現在、()は絶対必要です。「 $(2+3)-(2-2)$ 」を計算できないけれど、「 $((2+3)-(2-2))$ 」を計算できます。