

データ構造とアルゴリズム

課題 B 逆ポーランド記法

提出日: 2024/12/06

HI4 45 号 山口惺司

1. 課題

与えられた逆ポーランドの式を解く Python コードを作成せよ.

2. アルゴリズム

逆ポーランド記法とは：

- ・ $a + b$ を $ab+$ のように演算子を 2 個の変数の後ろに書く書き方.
- ・ $a + b * c$ は $abc*+$ となる.

アルゴリズム：

以下の手順に従って計算をする.

- ① 与えられた逆ポーランドの式を先頭から 1 文字ずつ見ていく.
- ② 変数が来たらスタックに積む.
- ③ 演算子が来たらスタックから 2 個降ろして演算を行い, 結果をスタックに積む.
- ④ 式が終わった時にスタックに残っている数が演算結果である.

フローチャート：

逆ポーランド記法のフローチャートを図 1 に示す.

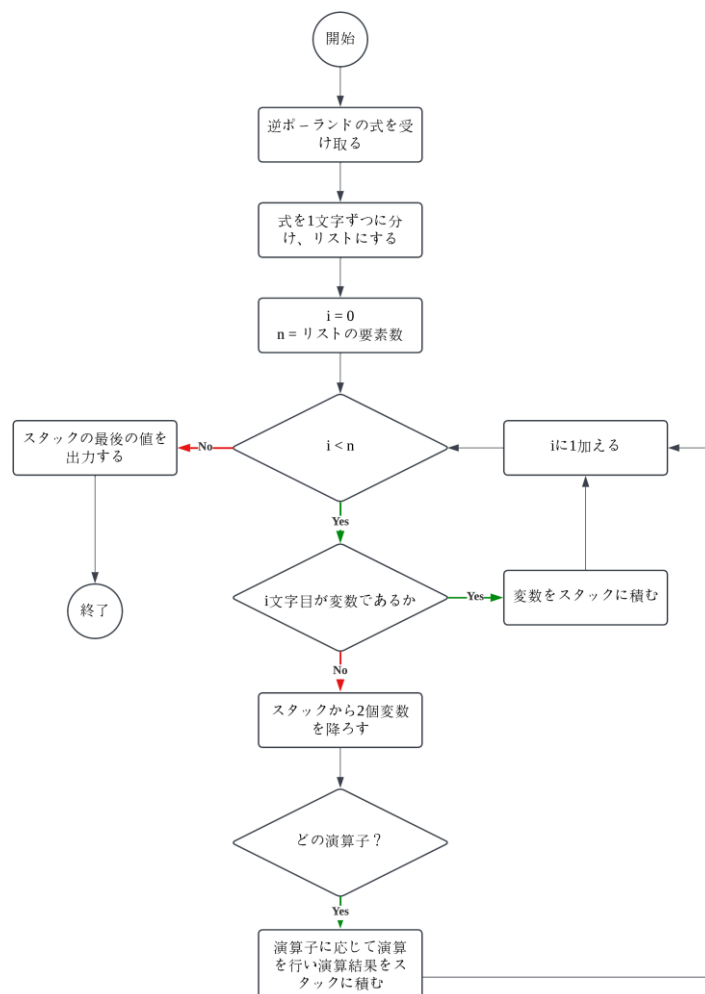


図 1 逆ポーランド記法のフローチャート

3. スクリプト

スクリプトを以下に示す.

また, 入力は必ず逆ポーランド記法の式であり, 予期せぬ記号などは入力されないものとする.

```
def rpn(tokens, data_list):
    stack = []

    for token in tokens:
        if token == '+':
            y = stack.pop()
            x = stack.pop()
            stack.append(x + y)
        elif token == '-':
            y = stack.pop()
            x = stack.pop()
            stack.append(x - y)
        elif token == '*':
            y = stack.pop()
            x = stack.pop()
            stack.append(x * y)
        elif token == '/':
            y = stack.pop()
            x = stack.pop()
            stack.append(x / y)
        else:
            for data in data_list[0]:
                if token in data:
                    index = data_list[0].index(data)
                    stack.append(float(data_list[1][index]))
                    break

    return stack.pop()

def main():
    equation = input("式を入力してください: ")
    tokens = equation.split()
    data_list = [[], []]
    for token in tokens:
        if token != '+' and token != '-' and token != '*' and
token != '/':
            print("変数に対応する値を入力してください. [" + token + "]:", end = "")
```

```

        value = input()
        data_list[0].append(token)
        data_list[1].append(value)

    result = rpn(tokens, data_list)
    print(f"計算結果: ", result)

if __name__ == "__main__":
    main()

```

4. 実行結果

表 1～4 の変数表に従って入力した時の実行結果をそれぞれ図 2～5 に示す.

表 1 実行結果 1 に対応する変数表

番号	変数名	数値
1	a	2
2	b	3
3	c	4

表 2 実行結果 2 に対応する変数表

番号	変数名	数値
1	a	8
2	b	4
3	c	5

表 3 実行結果 3 に対応する変数表

番号	変数名	数値
1	a	9
2	b	5
3	c	5
4	d	3

表 4 実行結果 4 に対応する変数表

番号	変数名	数値
1	a	-12
2	b	5
3	c	8
4	d	4

式を入力してください: a b c * +
変数に対応する値を入力してください. [a]: 2
変数に対応する値を入力してください. [b]: 3
変数に対応する値を入力してください. [c]: 4
計算結果: 14.0

図2 入力[a b c * +]の時の実行結果 1

式を入力してください: a b / c *
変数に対応する値を入力してください. [a]: 8
変数に対応する値を入力してください. [b]: 4
変数に対応する値を入力してください. [c]: 5
計算結果: 10.0

図3 入力[8 4 / 5 *] の時の実行結果 2

式を入力してください: a b c - + d /
変数に対応する値を入力してください. [a]: 9
変数に対応する値を入力してください. [b]: 5
変数に対応する値を入力してください. [c]: 5
変数に対応する値を入力してください. [d]: 3
計算結果: 3.0

図4 入力[9 5 5 - + 3]の時の実行結果 3

式を入力してください: a b c d / / /
変数に対応する値を入力してください. [a]: -12
変数に対応する値を入力してください. [b]: 5
変数に対応する値を入力してください. [c]: 8
変数に対応する値を入力してください. [d]: 4
計算結果: -4.8

図5 入力[-12 5 8 4] の時の実行結果 4

5. 考察

図2では「 $2 + 3 * 4$ 」, 図3では「 $8 / 4 * 5$ 」, 図4では「 $(9 + 5 - 5) / 3$ 」, 図5では「 $-12 / (5 / (8 / 4))$ 」を逆ポーランド記法で表しており, すべて計算結果が正しいので作成したプログラムは正しいと言える.