

問題3

1. アイデア

逆ポーランド記法で例えば $31 * 25 + 4 + *$ と入力された場合について考える。
 入力された $*$ を、文字列として配列 A に格納しておく。

	0	1	2	3	4	5	6	7	8
A	3	1	*	2	5	+	4	+	*

A[0] ~ A[8] まで順に見たとき、要素が ① 数字ならば別の配列 S に格納し、
 ② “+” ならば S に格納されている要素のうしろから 2 つを取り出し和を求め、さらに S に
 その和を格納し、③ “x” ならば S に格納されている要素のうしろから 2 つを取り出し
 積を求め、さらにその積を S に格納する ようなプログラムを作成すればよい。

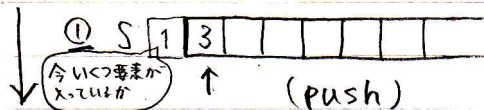
またこの際 ① 数字を S に格納する関数 (スタックの push) 、

② S のうしろ 2 つの和を求め S に格納する関数
 (スタックの pop) (スタックの push)

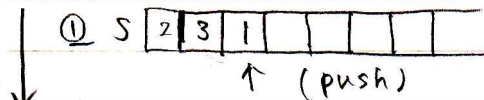
③ S のうしろ 2 つの積を求め S に格納する関数
 (スタックの pop) (スタックの push)。

と、① ~ ③ のどの関数を選択すればよいかを判断し、最後に計算結果を表示する
関数が必要となる。 (※ ④ 数字を S に格納する際に int 型にはおき関数も必要です)
 以上のイメージを、下に示す。

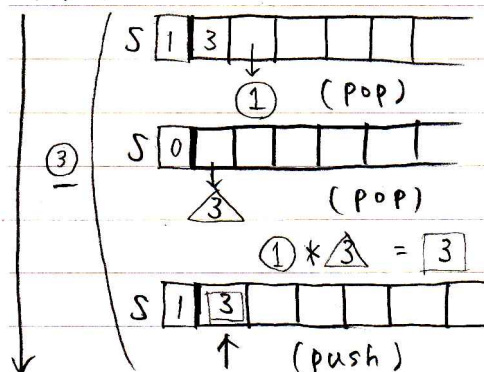
• A[0] = 3



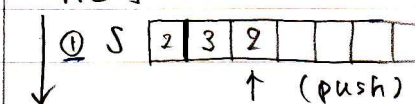
• A[1] = 1



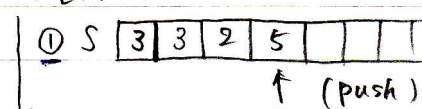
• A[2] = *



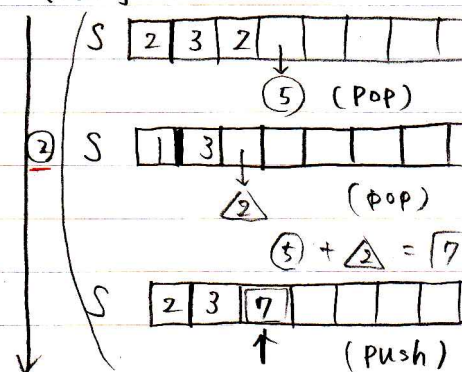
• A[3] = 2



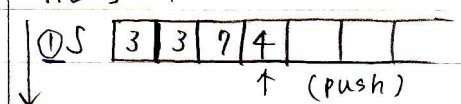
• A[4] = 5



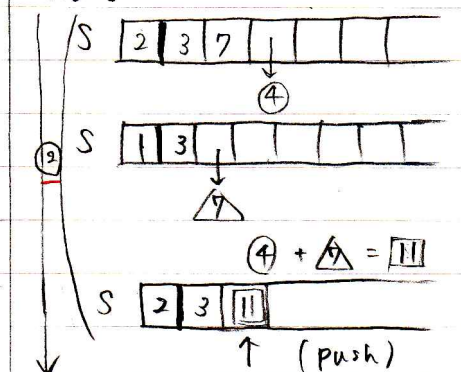
• A[5] = +



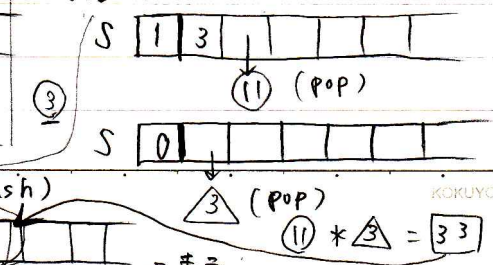
• A[6] = 4



• A[7] = +



• A[8] = *



表示

3. 実行結果

2
Z:¥c>gpo

31*25+4+*

33

Z:¥c>gpo

31+25*4*+

44

Z:¥c>gpo

83+4*55++33++8*

480

Z:¥c>gpo

83+++

スタックは空です。要素はありません。

スタックは空です。要素はありません。

11

4. 考察

今回作成したプログラムでは、gets 関数を用いてキーボードによる文字列の入力という形をとったので、入力できる数が 2ケタ以上のものは取り扱うことができない。

今後この問題を解決できるようなプログラムを作成しようと思うが、入力の際に gets ではなく配列に1つずつ要素を格納するような形をとれば良いが、作ることができなかった。

2