

1. Постфикс тэгшитгэл

Үйлдэл нь операндынхаа хойноо байдаг хэлбэрийг тэгшитгэл юм.

3 4 +

Уг хэлбэрийн тэгшитгэлийг бодох үйлдлийн дараалал гэж байхгүй (хаалт байхгүй, үржих хуваахыг түрүүлж хийхгүй). Урдаас нь эхлэн таарсан үйлдлээ шууд хийнэ.

3 - 2 * (22 - 4 + 8) + 2

3 2 22 4 - 8 + * - 2 +

гэж бичигдэх юм.

Хөрвүүлэх аргачлал

Үйлдлүүдэд зэрэглэл тогтооно.

1. +, -

2. *, /

гэж үзье.

Стак ашиглан $s[i]$ тэмдэгтийн хувьд дараах дүрмийн дагуу хөрвүүлэлтийг хийнэ. Тухайн тэмдэгтийг хэвлэвэл постфикс тэгшитгэлийн хэсэг болж байна гэж үзье.

1. Тоо бол шууд хэвлэнэ.

2. '(' бол такт шууд хийнэ.

3. Тэмдэгт бол

(a) ')' бол такт '(' хаалт хүртэлх бүгдийг гаргаж хэвлэнэ. Бас нээх хаалтыг хэвлэлгүйгээр гаргана.

(b) Стак хоосон эсвэл '(' хаалт хамгийн дээр байгаа эсвэл стакын хамгийн дээд талд байгаа үйлдэл, $s[i]$ -с бага зэрэглэлтэй бол такт хийнэ.

(c) Үгүй бол эрс бага зэрэглэлтэй үйлдэл стакын хамгийн дээд талд үлдэх эсвэл стак хоосон эсвэл нээх хаалт стакын дээд талд үлдтэл стакаас үйлдлүүдийг гаргаж хэвлэнэ. Дараагаар тухайн үйлдлийг такт хийнэ.

4. Бүх тэмдэгтүүдийг боловсруулж дууссан бол стакаас бүгдийг гарган хэвлэнэ.

Жишээ нь $3 - 2 * (22 - 4 + 8) + 2$ тэгшитгэлийн хувьд

1. 3 => 1-р дүрмээр шууд хэвлэгдэнэ.

stack: хоосон

гаралт: 3

2. - => 3.b дүрмээр такт оруулна.

stack: -

```
    гаралт: 3
3. 2 => 1-р дүрмээр шууд хэвлэгдэнэ.
    stack: -
    гаралт: 3 2
4. * => 3.b дүрмээр стакт оруулна.
    stack: - *
    гаралт: 3 2
5. ( => 2-р дүрмээр стакт оруулна.
    stack: - * (
    гаралт: 3 2
6. 22 => 1-р дүрмээр шууд хэвлэгдэнэ.
    stack: - * (
    гаралт: 3 2 22
7. - => 3.b дүрмээр стакт оруулна.
    stack: - * ( -
    гаралт: 3 2 22
8. 4 => 1-р дүрмээр шууд хэвлэгдэнэ.
    stack: - * ( -
    гаралт: 3 2 22 4
9. + => 3.c-р дүрмээр стакаас үйлдлүүдийг гаргаж, оруулна.
    stack: - * ( +
    гаралт: 3 2 22 4 -
10. 8 => 1-р дүрмээр шууд хэвлэгдэнэ.
    stack: - * ( +
    гаралт: 3 2 22 4 - 8
11. ) => 3.a дүрмээр стакаас гаргана.
    stack: - *
    гаралт: 3 2 22 4 - 8 +
12. + => 3.c дүрмээр стакаас үйлдлүүдийг гаргаж, оруулна.
    stack: +
    гаралт: 3 2 22 4 - 8 + * -
13. 2 => 1-р дүрмээр шууд хэвлэгдэнэ.
    stack: +
    гаралт: 3 2 22 4 - 8 + * - 2
14. 4-р дүрмээр бүгдийг стакаас гаргаж хэвлэнэ.
    stack:
    гаралт: 3 2 22 4 - 8 + * - 2 +
```

Даалгавар

postfix.c файл доторх кодыг гүйцээ. Ингэхдээ өөрсдийн бичсэн стак, жагсаалт бүтцийн кодыг ашиглана. Уг програм нь хоосон зайгаар тусгаарлан оруулсан тэгшитгэлийг постфикс тэгшитгэлд хөрвүүлээд. Үр дүнг нь бодож гаргана.

Задлах

Тэмдэгтэн цуваанаас тоо, тэмдэгтүүдийг ялган хөрвүүлэх. `tokenize(const char s[], struct List *p_list)` функц энэ үйлдлийг хийж `struct Token` төрлийн объект хадгалах `p_list` жагсаалтад хадгална.

Жагсаалтын объектуудыг *flag* талбараар нь тухайн объект тоо байна уу эсвэл тэмдэгт байна уу гэдгийг нь ялгана.

Жишээ нь уг жагсаалтын хамгийн эхний элемент нь дараах өгөгдөл байна.

flag	val	op
1	3	?

Хоёр дахь элемент нь

flag	val	op
0	?	'-'

өгөгдөл байна.

Постфиксрүү хөрвүүлэх

`void convert_to_postfix(struct List *p_tk, struct List *p_pf)` функц нь `p_tk` жагсаалтад байгаа объектуудыг постфикст хөрвүүлэн `p_pf` жагсаалтад хадгална.

Бодох

`int solve(struct List *p_pf)` функц нь постфикс тэгшитгэлийг бодно.

2. Стакаар рекурсивыг орлуулах

Бүтэн хоёртын модны даалгаврын pre-order, in-order, post-order үйлдлүүдийг стек ашиглан хийнэ үү.