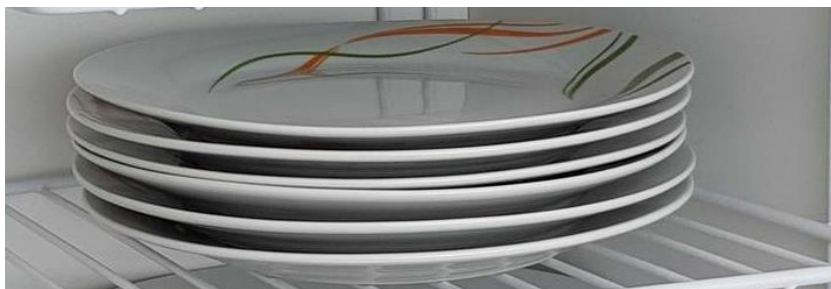


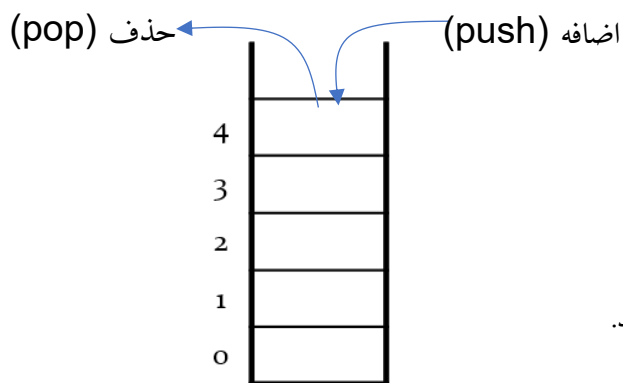
پشته (stack):

پشته یک نوع داده مجرد (ADT) است که در اکثر زبان‌های برنامه‌نویسی کاربرد رایجی دارد. دلیل این که این نوع داده، پشته نامیده شده، این است که از نظر ظاهری شبیه پشته است، یعنی به یک دسته از بشقاب‌های روی هم شباهت دارد.



در دنیای واقعی شما تنها می‌توانید بشقاب‌های جدید را روی دسته بشقاب‌ها بگذارید یا از روی آن بردارید. به طور مشابه ADT پشته نیز تنها از یک سمت امکان انجام عملیات‌ها بر روی داده‌ها را فراهم می‌کند. ما در هر زمان تنها به عناصر فوقانی پشته دسترسی داریم.

این ویژگی باعث می‌شود که پشته یک ساختار داده LIFO (Last-in-first-out) باشد یعنی ورودی-آخر-خروجی-اول. در پشته عملیات درج به نام عملیات PUSH نامیده و عملیات حذف به نام POP خوانده می‌شود. ساختمان داده پشته می‌تواند بوسیله ی آرایه ها و لیست پیوندی نمایش یابد. پشته می‌تواند دارای اندازه ثابت باشد و یا اندازه آن به طور دینامیک تغییر کند.



ADT پشته

مجموعه عناصر:

مجموعه ای از عناصر مرتب که از یک طرف (بالای پشته) قابل دستیابی اند.

عملیات اصلی:

creator: پشته خالی را ایجاد می‌کند.

isEmpty: خالی بودن پشته را بررسی می‌کند.

isFull: پر بودن پشته را بررسی می‌کند.

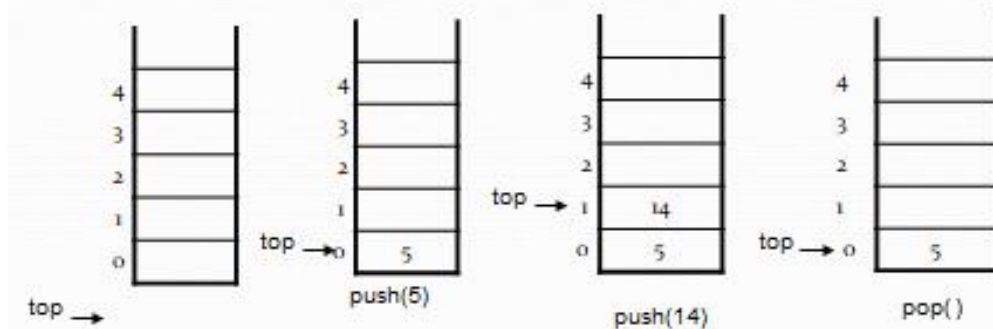
push: عنصری را به بالای پشته می‌افزاید.

pop: بالاترین عنصر پشته را حذف و ارائه می‌دهد.

peek: بالاترین عنصر پشته را بدون حذف آن ارائه می‌دهد.

ما باید یک متغیر برای مشخص کردن آخرین داده push شده به پشته در اختیار داشته باشیم. از آنجا که این اشاره گر همواره به عنصر فوقانی پشته اشاره می کند، نام آن top تعیین شده است. اشاره گر top بالاترین مقدار پشته را بدون حذف آن به دست می دهد.

top: به عنصر بالای پشته اشاره می کند. در ابتدا $top = -1$ است.



کلاس پیاده سازی پشته به زبان C++

```
#define SIZE 5
class stack{
private:
    int top;
    int a[SIZE];
public:
    stack();
    int isEmpty();
    int isFull();
    void push(int x);
    int pop();
    int peek();
    void display();
};
```

```
stack:: stack()
{
    top = -1;
}
```

تابع سازنده ی پشته :

یک پشته ی خالی ایجاد می کند.

بررسی خالی بودن پشته

اگر هیچ عنصری در پشته وجود نداشته باشد، این تابع مقدار 1 و در غیر اینصورت مقدار 0 را بر می گرداند.

```
int stack:: isEmpty()
{
    if( top== -1)
        return 1;
    return 0;
}
```

بررسی پر بودن پشته

اگر همه ی مکانهای پشته پر باشند، این تابع مقدار 1 و در غیر اینصورت مقدار 0 را بر می گرداند.

```
int stack:: isFull()
{
    if( top== SIZE-1)
        return 1;
    return 0;
}
```

عملیات push

قرار دادن یک عنصر جدید در پشته را push می گویند. این عملیات شامل مراحل زیر است:

- گام 1: بررسی کن که پشته پر است یا نه.
 - گام 2: اگر پشته پر باشد، اعلام خطا کرده و خارج شو.
 - گام 3: اگر پشته پر نبود، مقدار top را یک واحد افزایش بده تا به فضای خالی بعدی اشاره کند.
 - گام 4: عنصر داده مد نظر را در مکانی که top به آن اشاره می کند، اضافه کن.
- ```
void stack:: push(int x)
{
 if (isFull())
 { cout<<"stack is full";
 exit(); }
 else
 a[++top] = x;
}
```

## عملیات pop

دسترسی به محتوای بالای پشته در هنگام حذف آن را pop می‌گویند. در پیاده‌سازی پشته با استفاده از آرایه، عنصر داده‌ای واقعاً حذف نمی‌شود، بلکه تنها مقدار top یک واحد کاهش می‌یابد. اما در پیاده‌سازی پشته با استفاده از لیست پیوندی، واقعاً عنصر داده‌ای از فضای تخصیص یافته پشته پاک می‌گردد. یک عملیات pop می‌تواند شامل گام‌های زیر باشد:

- گام 1: بررسی کن که پشته خالی است یا نه؟
- گام 2: اگر پشته خالی بود، یک خطا اعلام کن و خارج شو.
- گام 3: اگر پشته خالی نبود، عنصر داده‌ای را که top مورد اشاره قرار می‌دهد برگردان.
- گام 4: مقدار top را یک واحد کم کن.

```
int stack:: pop()
```

```
{
 if (isEmpty ())
 { cout<<"stack is empty";
 exit(); }
 else
 return a[top - -];
}
```

## عملیات peek

دسترسی به محتوای بالای پشته بدون حذف آن را peek می‌گویند. این عملیات می‌تواند شامل گام‌های زیر باشد:

- گام 1: بررسی کن که پشته خالی است یا نه؟
- گام 2: اگر پشته خالی بود، یک خطا اعلام کن و خارج شو.
- گام 3: اگر پشته خالی نبود، عنصر داده‌ای را که top مورد اشاره قرار می‌دهد برگردان .

```
int stack:: peek()
```

```
{
 if (isEmpty())
 { cout<<"stack is empty";
 exit(); }
 else
 return a[top] ;
}
```

**مثال:** فرض کنید یک پشته داشته باشیم که به ترتیب از پایین به بالا دارای عناصر 1، 2 و 3 باشد. وضعیت پشته را پس از انجام دستورات زیر از چپ به راست مشخص کنید.

$\text{pop}(A)$ : بالاترین عنصر موجود در پشته را حذف کرده و در متغیر A قرار می دهد.

$\text{push}(A)$ : مقدار موجود در متغیر را در پشته قرار می دهد.

$\text{pop}(x), \text{pop}(y), \text{push}(x), \text{pop}(z), \text{pop}(x), \text{push}(y), \text{push}(z)$

| $\text{pop}(x)$ | $\text{pop}(y)$ | $\text{push}(x)$ | $\text{pop}(z)$ | $\text{pop}(x)$ | $\text{push}(y)$ | $\text{push}(z)$ |
|-----------------|-----------------|------------------|-----------------|-----------------|------------------|------------------|
|                 |                 |                  |                 |                 |                  |                  |
|                 |                 |                  |                 |                 |                  |                  |
| 2               |                 | 3                |                 |                 |                  | 3                |
| 1               | 1               | 1                | 1               |                 | 2                | 2                |
| $x=3$           | $x=3$           | $x=3$            | $x=3$           | $x=1$           |                  |                  |
| $y$             | $y=2$           | $y=2$            | $y=2$           | $y=2$           |                  |                  |
| $z$             |                 | $z$              | $z=3$           | $z=3$           |                  |                  |

مثال: مقدار نهایی متغیرهای A و B و C را پس از اجرای دستورات زیر از چپ به راست مشخص کنید.

$A=10 \quad B=2 \quad C=5$

$\text{push}(B), \text{push}(A+B), \text{pop}(C), \text{push}(A-B), \text{push}(C), \text{push}(B), \text{pop}(A), \text{pop}(B), \text{push}(A*B), \text{push}(C),$   
 $\text{pop}(A), \text{pop}(C), \text{pop}(B)$

| $\text{push}(B)$ | $\text{push}(A+B)$ | $\text{pop}(C)$ | $\text{push}(A-B)$ | $\text{push}(C)$ | $\text{push}(B)$ | $\text{pop}(A)$ | $\text{pop}(B)$ | $\text{Push}(A*B)$ |
|------------------|--------------------|-----------------|--------------------|------------------|------------------|-----------------|-----------------|--------------------|
|                  |                    |                 |                    |                  | 2                |                 |                 |                    |
|                  |                    |                 |                    | 12               | 12               | 12              |                 | 24                 |
|                  | 12                 |                 | 8                  | 8                | 8                | 8               | 8               | 8                  |
| 2                | 2                  | 2               | 2                  | 2                | 2                | 2               | 2               | 2                  |
| $A=10$           | $A=10$             | $A=10$          | $A=10$             | $A=10$           | $A=10$           | $A=2$           | $A=2$           | $A=2$              |
| $B=2$            | $B=2$              | $B=2$           | $B=2$              | $B=2$            | $B=2$            | $B=2$           | $B=12$          | $B=12$             |
| $C=5$            | $C=5$              | $C=12$          | $C=12$             | $C=12$           | $C=12$           | $C=12$          | $C=12$          | $C=12$             |

| $\text{push}(C)$ | $\text{pop}(A)$ | $\text{pop}(C)$ | $\text{pop}(B)$ |
|------------------|-----------------|-----------------|-----------------|
| 12               |                 |                 |                 |
| 24               | 24              |                 |                 |
| 8                | 8               | 8               |                 |
| 2                | 2               | 2               | 2               |
| $A=2$            | $A=12$          | $A=12$          | $A=12$          |
| $B=12$           | $B=12$          | $B=12$          | $B=8$           |
| $C=12$           | $C=12$          | $C=24$          | $C=24$          |

مثال: یک پشته ی خالی با گنجایش 5 عنصر را در نظر بگیرید. اگر اجازه داشته باشیم که اعداد 1 تا 5 را به ترتیب در یک پشته اضافه نماییم، کدام یک از ترتیبهای خروجی زیر امکان پذیر نیست؟ (اعداد خروجی، اعدادی هستند که به ترتیب از پشته با استفاده از عملیات pop خارج شده اند، آنها را از چپ به راست بخوانید)

الف) 1, 2, 3, 5, 4

ب) 1, 2, 3, 4, 5

ج) 3, 2, 4, 1, 5

بررسی حالت الف

| push(1) | push(2) | push(3) | push(4) | pop() | pop() | push(5) | pop() | pop() | Pop() |
|---------|---------|---------|---------|-------|-------|---------|-------|-------|-------|
|         |         |         |         |       |       |         |       |       |       |
|         |         |         | 4       |       |       |         |       |       |       |
|         |         | 3       | 3       | 3     |       | 5       |       |       |       |
|         | 2       | 2       | 2       | 2     | 2     | 2       | 2     |       |       |
| 1       | 1       | 1       | 1       | 1     | 1     | 1       | 1     | 1     |       |

بررسی حالت ب

| push(1) | push(2) | push(3) | pop() | push(4) | pop() | push(5) | pop() | pop() | Pop() |
|---------|---------|---------|-------|---------|-------|---------|-------|-------|-------|
|         |         |         |       |         |       |         |       |       |       |
|         |         |         |       |         |       |         |       |       |       |
|         |         | 3       |       | 4       |       | 5       |       |       |       |
|         | 2       | 2       | 2     | 2       | 2     | 2       | 2     |       |       |
| 1       | 1       | 1       | 1     | 1       | 1     | 1       | 1     | 1     |       |



بررسی حالت ج

| push(1) | pop() | push(2) | push(3) | push(4) | pop() | push(5) | pop() | pop() | Pop() |
|---------|-------|---------|---------|---------|-------|---------|-------|-------|-------|
|         |       |         |         |         |       |         |       |       |       |
|         |       |         |         | 4       |       | 5       |       |       |       |
|         |       |         | 3       | 3       | 3     | 3       | 3     |       |       |
| 1       |       | 2       | 2       | 2       | 2     | 2       | 2     | 2     |       |

تمرین: یک پشته ی خالی با گنجایش 5 عنصر را در نظر بگیرید. اگر اجازه داشته باشیم که اعداد 1 تا 5 را به ترتیب در یک پشته اضافه نماییم، کدام یک از ترتیبهای خروجی زیر امکان پذیر است؟ (اعداد خروجی، اعدادی هستند که به ترتیب از پشته با استفاده از عملیات pop خارج شده اند، آنها را از چپ به راست بخوانید)

الف) 1, 2, 5, 3, 4

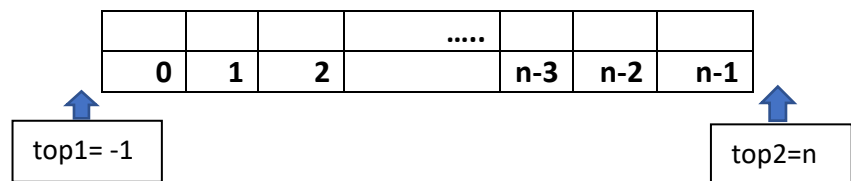
ب) 1, 2, 3, 4, 5

ج) 4, 1, 5, 3, 2

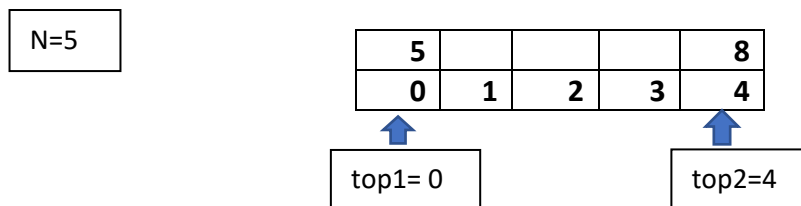
## پشته دو گانه

برای پیاده سازی دو پشته در یک آرایه نیاز به دو متغیر برای نشان دادن بالاترین عنصر هر پشته می باشد. بالاترین عنصر پشته اول با متغیر  $top1$  و بالاترین عنصر پشته دوم با متغیر  $top2$  مشخص می گردد و این دو پشته در جهت عکس یکدیگر حرکت می کنند.

مقدار اولیه  $top1 = -1$  و مقدار اولیه ی  $top2 = n$  می باشد.



اگر عنصری به پشته اول اضافه گردد  $top1 = top1 + 1$  و اگر عنصری به پشته دوم اضافه گردد  $top2 = top2 - 1$



شرط پر بودن پشته دو گانه

$$top2 = top1 + 1$$

## اولویت عملگر

بطور کلی اولویت عملگرها به صورت زیر می باشد:

1. ( ) ++ و- پیشوندی

2. عملگر منطقی not - (منفی) توان

3. عملگر منطقی and \* /

4. عملگر منطقی or + -

5. ++ و- پسوندی

بین عملگرهایی که اولویت مساوی دارند عملگری زودتر محاسبه می گردد که سمت چپ باشد.

### روش نمایش عبارات محاسباتی

میانوندی infix  $a + b$

پسوندی postfix  $ab+$

پیشوندی prefix  $+ab$

### مثال:

$$6 * 5 - 17 + 30 / 2 / 3 + 7$$

$$(6 * 5) - 17 + (30 / 2) / 3 + 7$$

$$\left( \left( (6 * 5) - 17 \right) + \left( (30 / 2) / 3 \right) \right) + 7$$

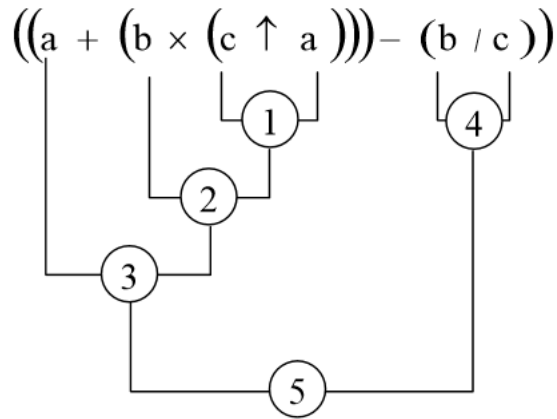
تمرین:

$$(9 + 15) / 3 * (4 + 2) * 7$$

### تبدیل عبارات میانوندی به پسوندی و پیشوندی بدون استفاده از پشته

1. پرانتز گذاری
2. برای تبدیل به پیشوندی، عملگر درون هر پرانتز را به سمت چپ منتقل می کنیم.
3. برای تبدیل به پسوندی، عملگر درون هر پرانتز را به سمت راست منتقل می کنیم.
4. پرانتزها را حذف می کنیم.



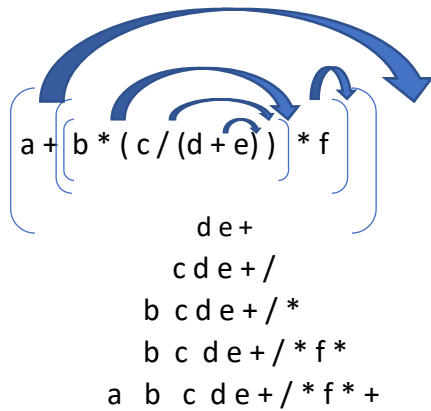


postfix =  $(a(b(ca))^{\wedge} \times + (bc) /) - = abca^{\wedge} \times + bc / -$

prefix =  $- + a \times b^{\wedge} ca / bc$

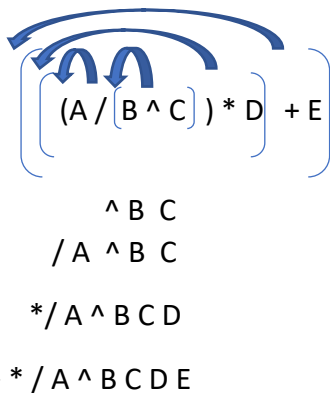
مثال :

$$a + b * (c / (d + e)) * f$$



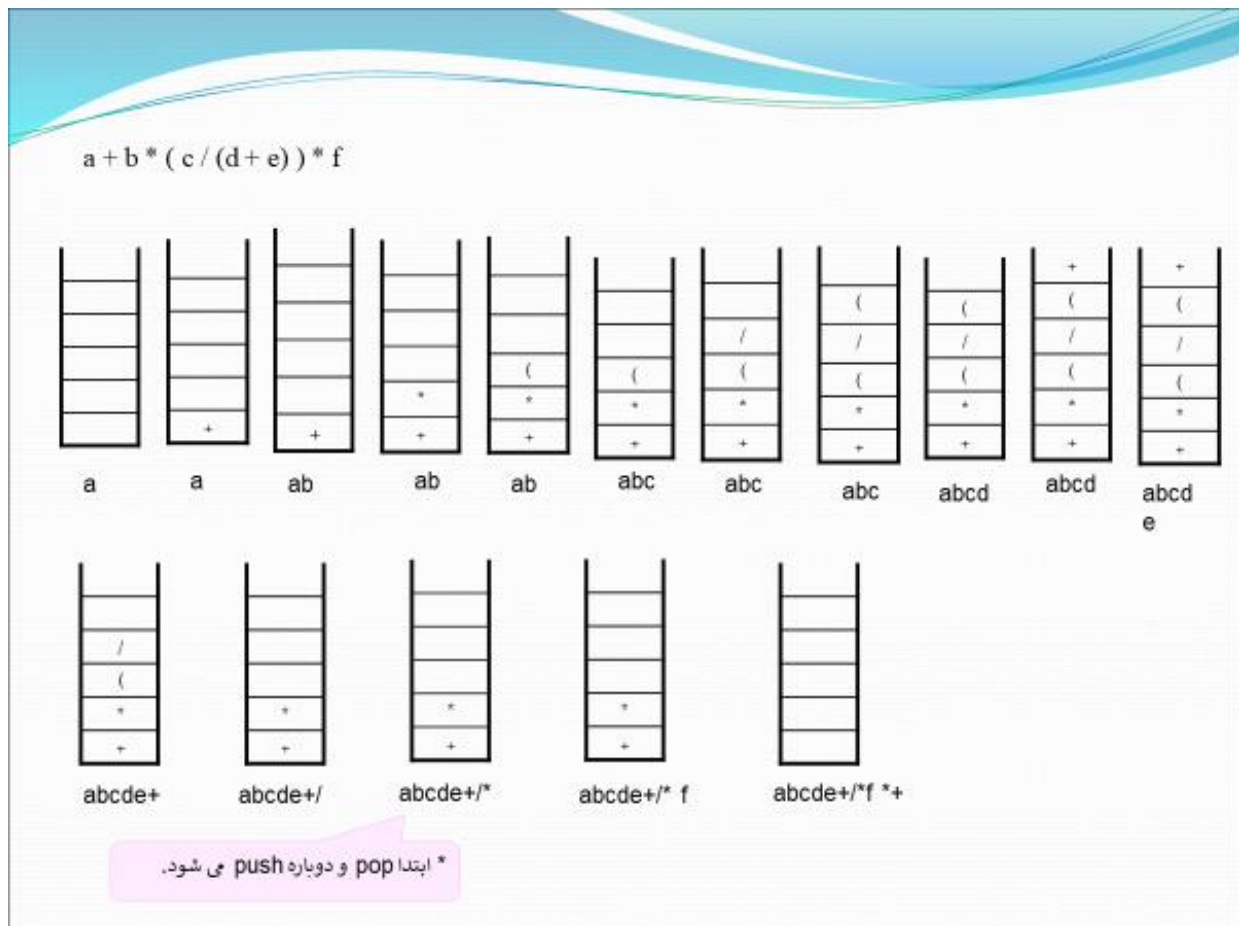
تمرین: عبارت میانوندی زیر را با استفاده از پرائتز گذاری به پیشوندی تبدیل نمایید.

$$(A / B ^ C) * D + E$$



## استفاده از پشته در تبدیل عبارات infix به postfix

1. عبارت را از چپ به راست پیمایش می کنیم.
2. پرانتز باز را در پشته push می کنیم.
3. عملوندها را در خروجی می نویسیم.
4. در صورتیکه به یک عملگر رسیدیم اگر عملگری که top پشته به آن اشاره می کند، عملگری با اولویت بیشتر یا مساوی نبود آنرا push می کنیم در غیر اینصورت عملگری که top پشته به آن اشاره می کند را pop کرده و در خروجی می نویسیم.
5. هرگاه به پرانتز بسته رسیدیم، آنقدر pop می کنیم تا به اولین پرانتز باز برسیم.



$$\left( \left( a + \left( b * (c \wedge a) \right) \right) - (b / c) \right)$$

|          |                                    |
|----------|------------------------------------|
| $\wedge$ | <input type="text" value="X"/>     |
| $($      | <input type="text" value="X"/>     |
| $*$      | <input type="text" value="X"/>     |
| $($      | <input type="text" value="X"/> $/$ |
| $+$      | <input type="text" value="X"/> $($ |
| $($      | <input type="text" value="X"/> $-$ |
| $($      |                                    |

تمرین: عبارت میانوندی زیر را با استفاده از پشته به عبارت پسوندی تبدیل نمایید.

### استفاده از یشته در تبدیل عبارات infix به prefix

مثال:

$$a + (b * c) ^ d / a - c * b$$

|   |                                     |   |                                     |   |                                     |                                     |   |                                     |  |                                     |  |
|---|-------------------------------------|---|-------------------------------------|---|-------------------------------------|-------------------------------------|---|-------------------------------------|--|-------------------------------------|--|
|   |                                     |   |                                     |   |                                     |                                     |   |                                     |  |                                     |  |
|   |                                     |   |                                     |   |                                     |                                     |   |                                     |  |                                     |  |
| * | <input checked="" type="checkbox"/> |   |                                     |   |                                     |                                     |   |                                     |  |                                     |  |
| ( | <input checked="" type="checkbox"/> | ^ | <input checked="" type="checkbox"/> | / | <input checked="" type="checkbox"/> |                                     | * | <input checked="" type="checkbox"/> |  |                                     |  |
| + | +                                   | + | +                                   | + | +                                   | <input checked="" type="checkbox"/> | - | -                                   |  | <input checked="" type="checkbox"/> |  |

|   |                                     |     |                                     |       |         |           |           |              |     |
|---|-------------------------------------|-----|-------------------------------------|-------|---------|-----------|-----------|--------------|-----|
|   |                                     |     |                                     |       |         |           |           |              |     |
|   |                                     |     |                                     |       |         |           |           |              |     |
| c | <input checked="" type="checkbox"/> | d   | <input checked="" type="checkbox"/> | a     | b       |           |           |              |     |
| b | <input checked="" type="checkbox"/> | *bc | <input checked="" type="checkbox"/> | ^*bcd | /^*bcda |           |           | c            | *cb |
| a | a                                   | a   | a                                   | a     | a       | +a/^*bcda | +a/^*bcda | -a/^*bcda*cd |     |

$$a + b * c ^ { ( 2 - b ) } * c / ( d + a )$$

|   |                                             |                      |                      |   |                      |                        |   |                      |                      |  |
|---|---------------------------------------------|----------------------|----------------------|---|----------------------|------------------------|---|----------------------|----------------------|--|
| - | <input type="text"/>                        |                      |                      |   |                      |                        |   |                      |                      |  |
| ( | <input type="text"/> + <input type="text"/> |                      |                      |   |                      |                        |   |                      |                      |  |
| ^ | ^                                           | <input type="text"/> |                      |   |                      | ( <input type="text"/> |   |                      |                      |  |
| * | *                                           | *                    | <input type="text"/> | * | <input type="text"/> | /                      | / | <input type="text"/> |                      |  |
| + | +                                           | +                    | +                    | + | +                    | +                      | + | +                    | <input type="text"/> |  |

|   |     |               |                  |                    |                     |                        |                          |
|---|-----|---------------|------------------|--------------------|---------------------|------------------------|--------------------------|
| b |     |               |                  |                    |                     |                        |                          |
| 2 | -2b |               |                  | a                  |                     |                        |                          |
| c | c   | $\wedge c-2b$ | c                | d                  | +da                 |                        |                          |
| b | b   | b             | *b $\wedge c-2b$ | **b $\wedge c-2bc$ | ***b $\wedge c-2bc$ | /**b $\wedge c-2bc+da$ |                          |
| a | a   | a             | a                | a                  | a                   | a                      | +a/**b $\wedge c-2bc+da$ |

با استفاده از یک پشته می توان عبارت postfix را به infix تبدیل کرد. برای این منظور رشته را از چپ پردازش می کنیم . هر عملوند درون پشته push می شود . با رسیدن به هر عملگر، دو عنصر پشته pop شده و بصورت infix نوشته می شود . سپس عبارت infix تولید شده درون پشته push می شود . در پایان پردازش رشته ورودی، پشته حاوی یک عنصر است که شکل infix مورد نظر می باشد . خروجی infix لزوماً باید برانترگذاری شده باشد. عملوند top پشته سمت راست عملگر نوشته می شود.

$$abca^* + bc/-$$

|   |                                     |         |                                     |              |                                     |                  |                                     |                                     |                                     |
|---|-------------------------------------|---------|-------------------------------------|--------------|-------------------------------------|------------------|-------------------------------------|-------------------------------------|-------------------------------------|
| a | <input checked="" type="checkbox"/> |         |                                     |              |                                     |                  |                                     |                                     |                                     |
| c | <input checked="" type="checkbox"/> | $(c^a)$ | <input checked="" type="checkbox"/> | c            |                                     |                  |                                     | <input checked="" type="checkbox"/> |                                     |
| b | b                                   | b       | <input checked="" type="checkbox"/> | $(b^*(c^a))$ | <input checked="" type="checkbox"/> | b                | <input checked="" type="checkbox"/> | $(b/c)$                             | <input checked="" type="checkbox"/> |
| a | a                                   | a       | a                                   | a            | <input checked="" type="checkbox"/> | $(a+(b^*(c^a)))$ | $(a+(b^*(c^a)))$                    | $(a+(b^*(c^a)))$                    | <input checked="" type="checkbox"/> |

$$((a+(b*(c^a))))-(b/c))$$

## تبدیل عبارت infix به prefix

با استفاده از یک پشته می توان عبارت prefix را به infix تبدیل کرد. برای این منظور رشته را از راست پردازش می کنیم . هر عملوند درون پشته push می شود . با رسیدن به هر عملگر، دو عنصر پشته pop شده و بصورت infix نوشته می شود . سپس عبارت infix تولید شده درون پشته push می شود . در پایان پردازش رشته ورودی، پشته حاوی یک عنصر است که شکل infix مورد نظر می باشد . خروجی infix لزوماً باید پراترگذاری شده باشد. عملوند top پشته سمت چپ عملگر نوشته می شود.

مثال:

$$- +a*b ^ ca/bc$$

|   |                                     |                                     |                                     |                                     |                                     |                                     |                                     |               |                                     |
|---|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|-------------------------------------|---------------|-------------------------------------|
|   |                                     |                                     |                                     |                                     |                                     |                                     |                                     |               |                                     |
|   |                                     |                                     |                                     |                                     |                                     |                                     |                                     |               |                                     |
|   | c                                   | <input checked="" type="checkbox"/> | b                                   | <input checked="" type="checkbox"/> | a                                   | <input checked="" type="checkbox"/> |                                     |               |                                     |
| b | <input checked="" type="checkbox"/> | a                                   | <input checked="" type="checkbox"/> | (c^a)                               | <input checked="" type="checkbox"/> | (b*(c^a))                           | <input checked="" type="checkbox"/> | (a+(b*(c^a))) | <input checked="" type="checkbox"/> |
| c | <input checked="" type="checkbox"/> | (b/c)                               | (b/c)                               | (b/c)                               | (b/c)                               | (b/c)                               | (b/c)                               | (b/c)         | <input checked="" type="checkbox"/> |

$$((a+(b*(c^a)))-(b/c))$$

تمرین: عبارات زیر را به پیشوندی تبدیل کنید.

$$abca^*+bc/-$$

$$abcd-/ +ab-cd5^*+abc+/-/-$$