

به نام خدا

ساختمان داده ها و الگوریتم ها

محمد مهدی گیلانیان صادقی

دانشگاه آزاد اسلامی واحد قزوین

نیمسال دوم ۱۴۰۱ - ۱۴۰۲



• کاربردهای پشته:

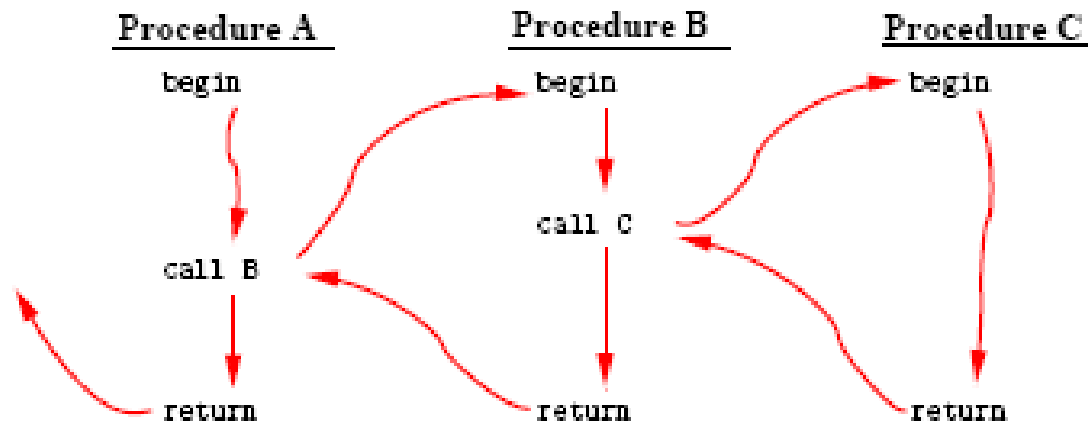
(۱) در فراخوانی توابع

(۲) در ارزیابی عبارات



کاربرد پشته در فراخوانی توابع:

- وقتی یک تابع فراخوانی می شود یک رکورد فعالیت برای آن ایجاد می شود.
- اگر تابع F1 در تابع F2 فراخوانی شود ادامه اجرای تابع F2 تا زمان اتمام اجرای F1 به تعویق می افتد.
- پشته ساختمان داده مناسبی برای ذخیره رکورد فعالیت های توابع است.



انتقال کنترل برنامه در فراخوانی و بازگشت



return n*fact(n-1)

```
long int fact(int n) {
    int x,y;
    if (n== 0)
        return 1;
    x = n-1;
    y=fact(x);
    return n*y;
}
```

کاربرد پشته در فراخوانی توابع (۲):

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |

n x y
a) Init.

| | | |
|---|---|---|
| | | |
| | | |
| | | |
| 4 | - | - |

n x y
b) fact (4)

| | | |
|---|---|---|
| | | |
| | | |
| 3 | - | - |
| 4 | 3 | - |

n x y
c) fact (3)

| | | |
|---|---|---|
| | | |
| 2 | - | - |
| 3 | 2 | - |
| 4 | 3 | - |

n x y
d) fact (2)

| | | |
|---|---|---|
| 1 | - | - |
| 2 | 1 | - |
| 3 | 2 | - |
| 4 | 3 | - |

n x y
e) fact (1)

| | | |
|---|---|---|
| 0 | - | - |
| 1 | 0 | - |
| 2 | 1 | - |
| 3 | 2 | - |
| 4 | 3 | - |

n x y
f) fact (0)

```
int main () {
    int n=4;
    cout << fact(n);
    return 0;
}
```

| | | |
|---|---|---|
| 1 | 0 | 1 |
| 2 | 1 | - |
| 3 | 2 | - |
| 4 | 3 | - |

n x y
g) y=fact(0)

| | | |
|---|---|---|
| 2 | 1 | 1 |
| 3 | 2 | - |
| 4 | 3 | - |

n x y
h) y=fact(1)

| | | |
|---|---|---|
| 3 | 2 | 2 |
| 4 | 3 | - |

n x y
i) y=fact(2)

| | | |
|---|---|---|
| 4 | 3 | 6 |
|---|---|---|

n x y
j) y=fact(3)

| | | |
|--|--|--|
| | | |
| | | |
| | | |
| | | |

n x y
k) cout<<fact(n)



کاربرد پشته در ارزیابی عبارات:

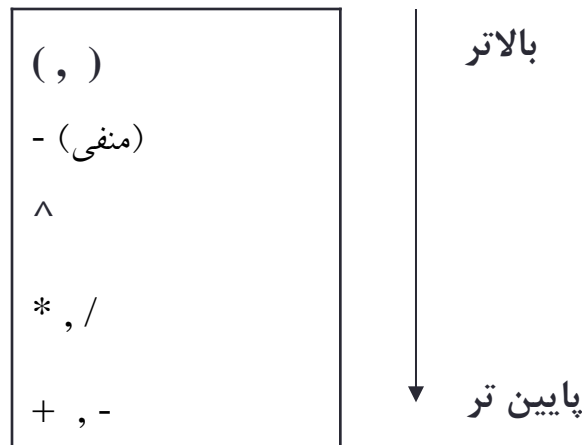
سه شکل نمایش عبارات:

$A+B$ عبارت میانوندی (infix)

$+AB$ عبارت پیشوندی (prefix)

$AB+$ عبارت پسوندی (postfix)

اولویت عملگرها :





■ تبدیل عبارت میانوندی به پسوندی یا پیشوندی:

(۱) با پرانتز گذاری

(۲) با پشته

■ تبدیل عبارت پسوندی به میانوندی

■ تبدیل عبارت پیشوندی به میانوندی



تبدیل عبارت میانوندی به پسوندی یا پیشوندی با پرانتزگذاری :

infix: $10 * 5 + (20 + 4) / 2 ^ 3$

infix: $10 * 5 + \underline{(20 + 4)} / 2 ^ 3$

infix: $10 * 5 + \underline{(20 + 4)} / \underline{(2 ^ 3)}$

infix: $\underline{(10 * 5)} + \underline{(20 + 4)} / \underline{(2 ^ 3)}$

infix: $\underline{(10 * 5)} + \underline{((20 + 4) / (2 ^ 3))}$

infix: $\underline{((10 * 5) + ((20 + 4) / (2 ^ 3)))}$

Postfix: $10 \ 5 \ * \ 20 \ 4 \ + \ 2 \ 3 \ ^ \ / \ +$

Prefix: $+ \ * \ 10 \ 5 \ / \ + \ 20 \ 4 \ ^ \ 2 \ 3$



تبدیل عبارت میانوندی به پسوندی با پشته :

✓ پشته ای خالی برای عملگرها ایجاد کنید.

✓ تا زمانی که خطایی رخ نداده و به انتهای عبارت میانوندی نرسیده اید اعمال زیر را انجام دهید. اگر نشانه:

● پرانتز باز است، آن را در پشته قرار دهید.

● پرانتز بسته است، عناصر پشته را آنقدر حذف کنید و در عبارت پسوندی قرار دهید تا به پرانتز باز برسید، ولی پرانتز باز را در عبارت پسوندی قرار ندهید.

● عملگر است، اگر پشته خالی است، یا تقدم این عملگر از تقدم عملگر بالای پشته بیشتر است آن را در پشته قرار دهید، وگرنه عنصر بالای پشته را حذف کنید و در عبارت پسوندی قرار دهید. سپس عملگر ورودی را با عملگر جدید موجود در پشته مقایسه کنید و این عمل را آنقدر ادامه دهید تا پشته خالی شود یا تقدم عملگر موجود در پشته کمتر از آن عملگر شود. در این صورت، آن عملگر را در پشته قرار دهید.

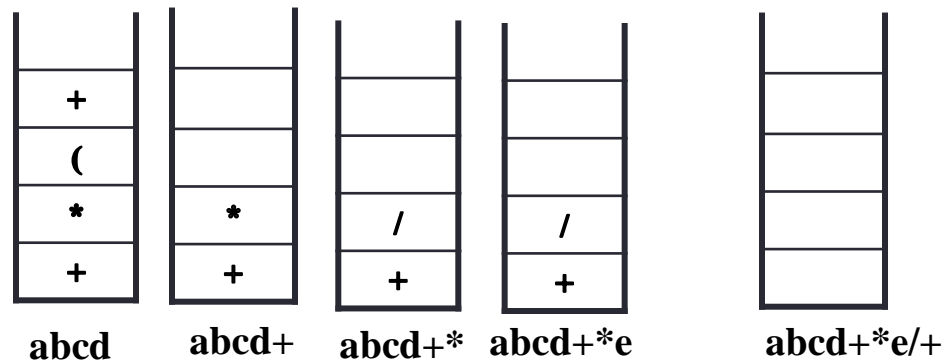
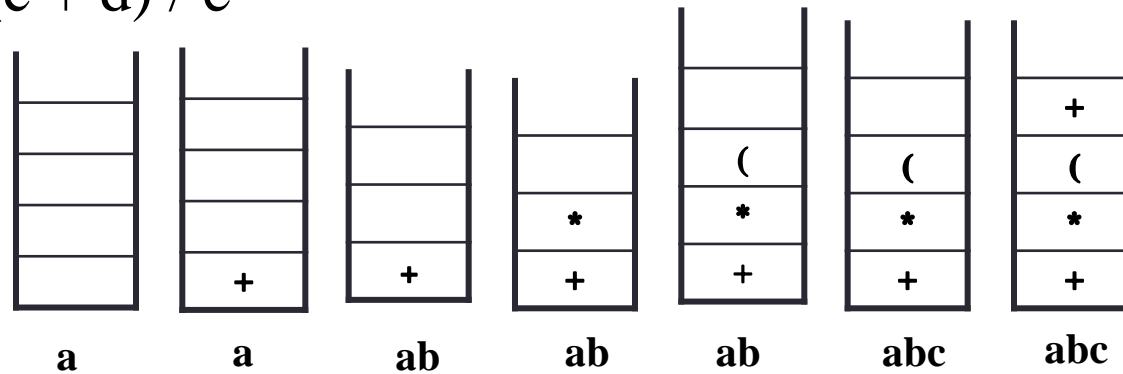
● عملوند است، آن را در عبارت پسوندی قرار دهید.

✓ وقتی به انتهای عبارت میانوندی رسیدی، عناصر موجود در پشته را حذف و در عبارت پسوندی قرار دهید تا پشته خالی شود.



تبدیل عبارت میانوندی به پسوندی با پشته (۲):

→
infix: $a + b * (c + d) / e$





ساختمان داده ها و الگوریتم ها

صفحه ۹

تبدیل عبارت پسوندی به میانوندی با پشته :

- عبارت را از چپ به راست بررسی می کنیم.
- عملوندها را در پشته قرار می دهیم و با رسیدن به هر عملگر دو عملوند خارج کرده و بعد از محاسبه حاصل عبارت را داخل پشته قرار می دهیم.

→
postfix: $abcd+*e/+$

$a=2, b=3, c=16, d=4, e=2$

| | | | | | | | | |
|---|---|---|---|-------|-----------|-----------|-------------|---------------|
| | | | d | | | | | |
| | | c | c | (c+d) | | e | | |
| | b | b | b | b | $b*(c+d)$ | $b*(c+d)$ | $b*(c+d)/e$ | |
| a | a | a | a | a | a | a | a | $a+b*(c+d)/e$ |

infix



ساختمان داده ها و الگوریتم ها

صفحه ۱۰

تبدیل عبارت میانوندی به پیشوندی با پشته :

- دو پشته ایجاد کنید.
- نشانه ها را از آخر عبارت میانوند جدا می کنیم. اگر نشانه :
 - ✓ پرائتز بسته بود یا عملگر بود در پشته شماره ۱ قرار می دهیم.
 - ✓ عملوند بود در پشته شماره ۲ قرار می دهیم.
 - ✓ پرائتز باز بود، از پشته شماره ۱ دو نشانه را بر می داریم و عملگر را در پشته شماره ۲ قرار می دهیم.
- در پایان هر چه در پشته شماره ۱ است برمی داریم و در عبارت پیشوندی قرار می دهیم.



ساختمان داده ها و الگوریتم ها

صفحه ۱۱

تبدیل عبارت میانوندی به پیشوندی با پشته :

Infix: $a+b*(c-d)/e$

| نشانه | پشته ۱ | پشته ۲ |
|-------|--------|-----------|
| e | | e |
| / | / | e |
|) | /) | e |
| d | /) | ed |
| - | /) - | ed |
| c | /) - | edc |
| (| / | edc- |
| * | / * | edc- |
| b | / * | edc-b |
| + | + | edc-b*/ |
| a | + | edc-b*/a |
| | | edc-b*/a+ |

pop

Prefix: $+ a / * b - c d e$

تبدیل عبارت پیشوندی به میانوندی با پشته :

- عبارت را از راست به چپ بررسی می کنیم.
- عملوندها را در پشته قرار می دهیم و با رسیدن به هر عملگر دو عملوند خارج کرده و بعد از محاسبه حاصل عبارت را داخل پشته قرار می دهیم.

Prefix: + a / * b - c d e a=2, b=3, c=16, d=4, e=2

| | | | | | | | | |
|---|---|---|-----|-----|---------|-----------|-----------|-------------|
| | | | | | | | | |
| | | | | | | | | |
| | | c | | b | | | | |
| | d | d | c-d | c-d | b*(c-d) | | a | |
| e | e | e | e | e | e | b*(c-d)/e | b*(c-d)/e | a+b*(c-d)/e |

infix



ساختمان داده ها و الگوریتم ها

صفحه ۱۳

تمرین ۱) عبارات infix زیر را با استفاده از پشته به postfix و prefix تبدیل کنید.

الف) $x + y * (a - b + c / d)^p / z - a * b$

ب) $a / b - c - d^e^f$

تمرین ۲) عبارات postfix زیر را استفاده از پشته به infix تبدیل کنید. (ارزیابی عبارت)

الف) $A B C * D / +$

ب) $a b c d e + / * f * +$

تمرین ۳) عبارات prefix زیر را استفاده از پشته به infix تبدیل کنید. (ارزیابی عبارت)

الف) $+ - * ^ A B C D / E / F + G H$

ب) $+ a / b * + d e - a c$



پایان