



# داده ساختار و الگوریتم

**Data Structures**

میلاذ سلطانی

# فهرست مطالب فصل دوم

- مقدمه
- تعریف پشته
- پیاده سازی پشته
- عملیات پشته
- درج
- حذف
- بازیابی
- کاربردهای پشته

## پشته‌ها

## Stack

# مقدمه

- پشته ساختمان داده ای است که عمل حذف کردن و درج عناصر در آن، از بالای پشته صورت می گیرد.
- عملکرد پشته به **آخرین ورودی = اولین خروجی** یا **LIFO** شهرت دارد. یعنی آخرین عنصری که وارد پشته می شود، اولین عنصری است که از آن خارج می شود.

مثال :

- ✓ چیدن کتاب ها روی هم
- ✓ گذاشتن چندین سکه روی هم
- ✓ چیدن بشقاب ها در کابیت

# پشته به عنوان یک ADT

- آرایه را می توان بصورت یک نوع داده انتزاعی در نظر گرفت :

## ADT پشته

مجموعه‌ای از عناصر داده‌ها:

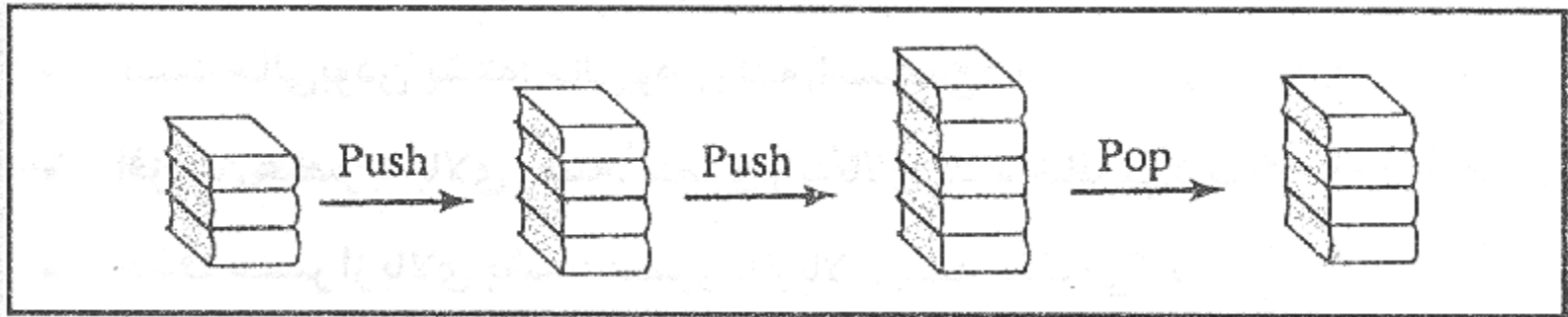
مجموعه‌ای از عناصر مرتب که از یک طرف قابل دستیابی اند و این طرف، بالای پشته نام دارد.

عملیات اصلی:

- ایجاد پشته: پشته خالی را ایجاد می‌کند.
- تست خالی بودن و پر بودن پشته: خالی بودن و پر بودن پشته را تست می‌کند.
- افزودن عنصر به بالای پشته: عنصری را به بالای پشته اضافه می‌کند.
- حذف عنصر از بالای پشته: عنصری را از بالای پشته حذف می‌کند.
- بازیابی عنصر بالای پشته: عنصری را از بالای پشته بازیابی می‌کند، ولی حذف نمی‌کند.

# پیاده سازی پشته

- جهت پیاده سازی پشته به یک قفسه کتاب فکر کنید که کتاب ها بصورت زیر در آن قرار گرفته اند



- برای پشته ای بصورت بالا دو مدل پیاده سازی داریم:

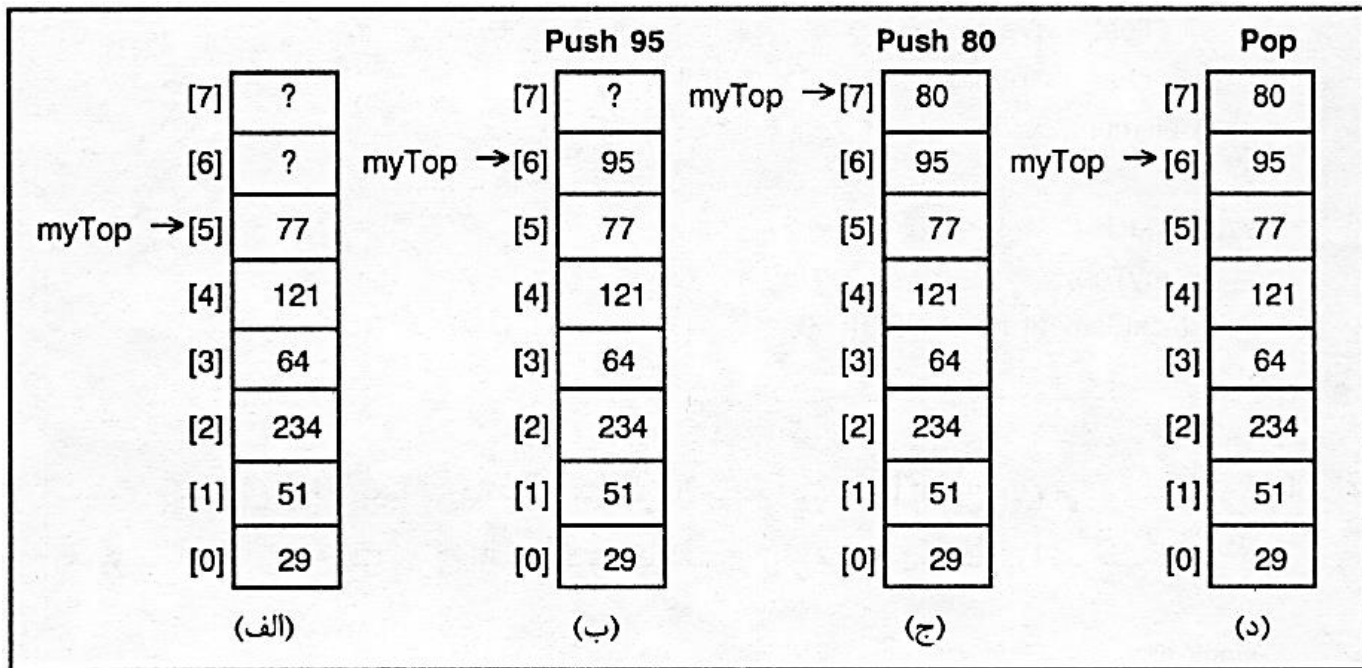
(۱) پیاده سازی خطی

(۲) پیاده سازی پیوندی

# پیاده سازی پشته با آرایه

- یک آرایه عناصر پشته را نگهداری می کند.
- یک متغیر به نام mytop برای نشان دادن آدرس عنصر ابتدای پشته دارد.
- عمل اضافه کردن و حذف کردن پشته از مکان mytop صورت می گیرد.

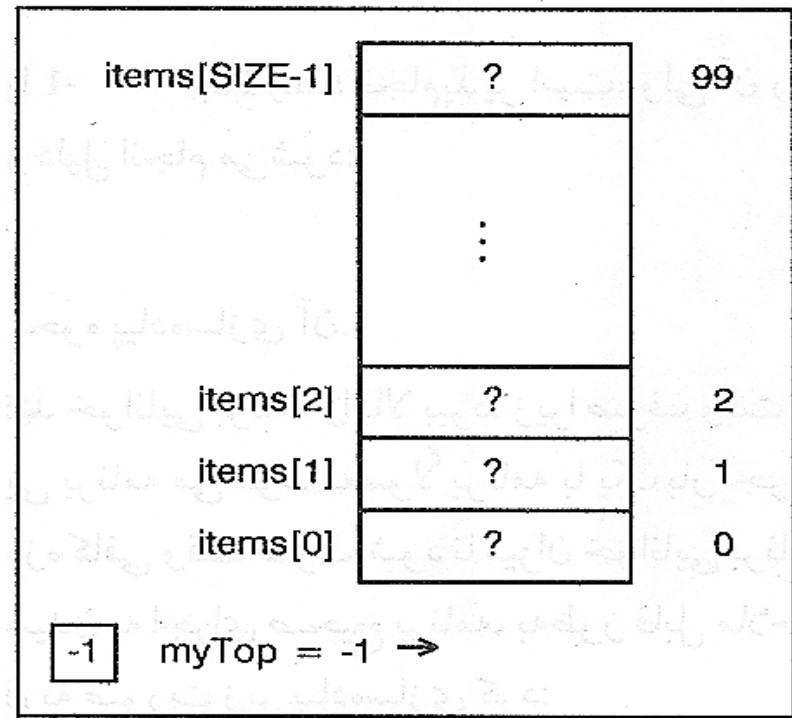
مثال:



# تعریف پشته با آرایه در زبان C

- در تعریف پشته باید یک آرایه و یک متغیر نشان دهنده بالای پشته تعریف شده و پشته خالی تعریف شود.

```
#define    SIZE    100
struct    stack {
    int myTop;
    int items[SIZE];
};
stack s;
s.myTop = -1;
```



تمونه‌ای از پشته با آرایه.

# پیاده سازی عمل تست پر بودن پشته

- برای درج در پشته بایستی پشته حداقل یک فضای خالی داشته باشد. لذا قبل از انجام عمل درج بایستی از پر نبودن پشته اطمینان حاصل شود.

```
int full (stack *s)
{
    if(s -> myTop == SIZE - 1)
        return 1; /* مقدار ۱ ارزش درستی دارد */
    else
        return 0; /* مقدار ۰ ارزش نادرستی دارد */
}
```

آیا پشته پر است ؟

بله پشته پر است.

خیر پشته پر نیست.

**نکته !** باید توجه داشت که پر نبودن پشته به معنی خالی بودن آن نیست، بلکه به این معنی است که حداقل یک فضای خالی بالای پشته وجود دارد.

# پیاده سازی عمل درج در پشته

- پس از اینکه از پر نبودن پشته اطمینان حاصل شد، آنگاه :
  ۱. متغیر نشان دهنده بالای پشته یک واحد افزایش می یابد.
  ۲. عنصر بالای پشته درج می شود.

```
void pushAndTest(stack *s, int x, int *overflow)
```

```
{
```

```
    if(full(s))
```

```
        *overflow = 1;
```

```
    else
```

```
{
```

```
    *overflow = 0;
```

```
    s -> items[++(s -> myTop)] = x;
```

```
}
```

```
}
```

✓ X مقداری است که در پشته درج

می شود.

✓ overflow اگر 1 باشد یعنی

پشته پر است و اگر 0 باشد یعنی

پشته پر نیست.

عنصر به بالای پشته اضافه می شود

# پیاده سازی عمل تست خالی بودن پشته

- برای حذف از پشته بایستی پشته حداقل یک عنصر داشته باشد. لذا قبل از انجام عمل حذف بایستی از خالی نبودن پشته اطمینان حاصل شود.

```
int empty(stack *s)
{
    if(s -> myTop == -1)
        return 1; /* مقدار ۱ ارزش درستی دارد */
    else
        return 0; /* مقدار ۰ ارزش نادرستی دارد */
}
```

آیا پشته خالی است ؟

بله پشته خالی است.

خیر پشته خالی نیست.

**نکته !** باید توجه داشت که خالی نبودن پشته به معنی پر بودن آن نیست، بلکه به این معنی است که حداقل یک عنصر درون پشته وجود دارد.

# پیاده سازی عمل حذف از پشته

- پس از اینکه از خالی نبودن پشته اطمینان حاصل شد، آنگاه :
  ۱. عنصر بالای پشته حذف می شود.
  ۲. متغیر نشان دهنده بالای پشته یک واحد کاهش می یابد.

```
void popAndTest(stack *s, int *x, int *underflow)
{
    if(empty(s))  بررسی خالی بودن پشته
    {
        *underflow = 1;
    }
    else  اگر پشته خالی نیست
    {
        *x = s -> items[(s -> myTop) --];
        *underflow = 0;
    }
}
```

 عنصر بالای پشته حذف می شود

✓ X مقداری است که از پشته حذف می شود.

✓ underflow اگر 1 باشد یعنی پشته خالی است و اگر 0 باشد یعنی پشته خالی نیست.

# پیاده سازی عمل بازیابی از پشته

- عمل بازیابی دقیقاً همان عمل حذف از پشته است ولی با این تفاوت که عنصری از پشته حذف نمی کند، فقط عنصر بالای پشته را نشان می دهد.

```
void topAndTest(stack *s, int *x, int *underflow)
```

```
{  
    if(empty(s))  
        *underflow = 1;  
    else  
    {  
        *x = s -> items[s -> myTop];  
        *underflow = 0;  
    }  
}
```

بررسی خالی بودن پشته

اگر پشته خالی نیست

دسترسی به عنصر بالای پشته

وقتی یک تابع در برنامه فراخوانی می شود :

۱. یک کپی از رکورد فعالیت آن در پشته قرار می گیرد.

۲. پارامترها ذخیره می شوند.

۳. کنترل به آدرس شروع بدنه تابع منتقل می شود.

رکورد فعالیت شامل موارد زیر است:

۱. پارامترها

۲. اطلاعات حالت فراخوان

۳. متغیرهای محلی

۴. حافظه های موقت برای انجام کارهای میانی

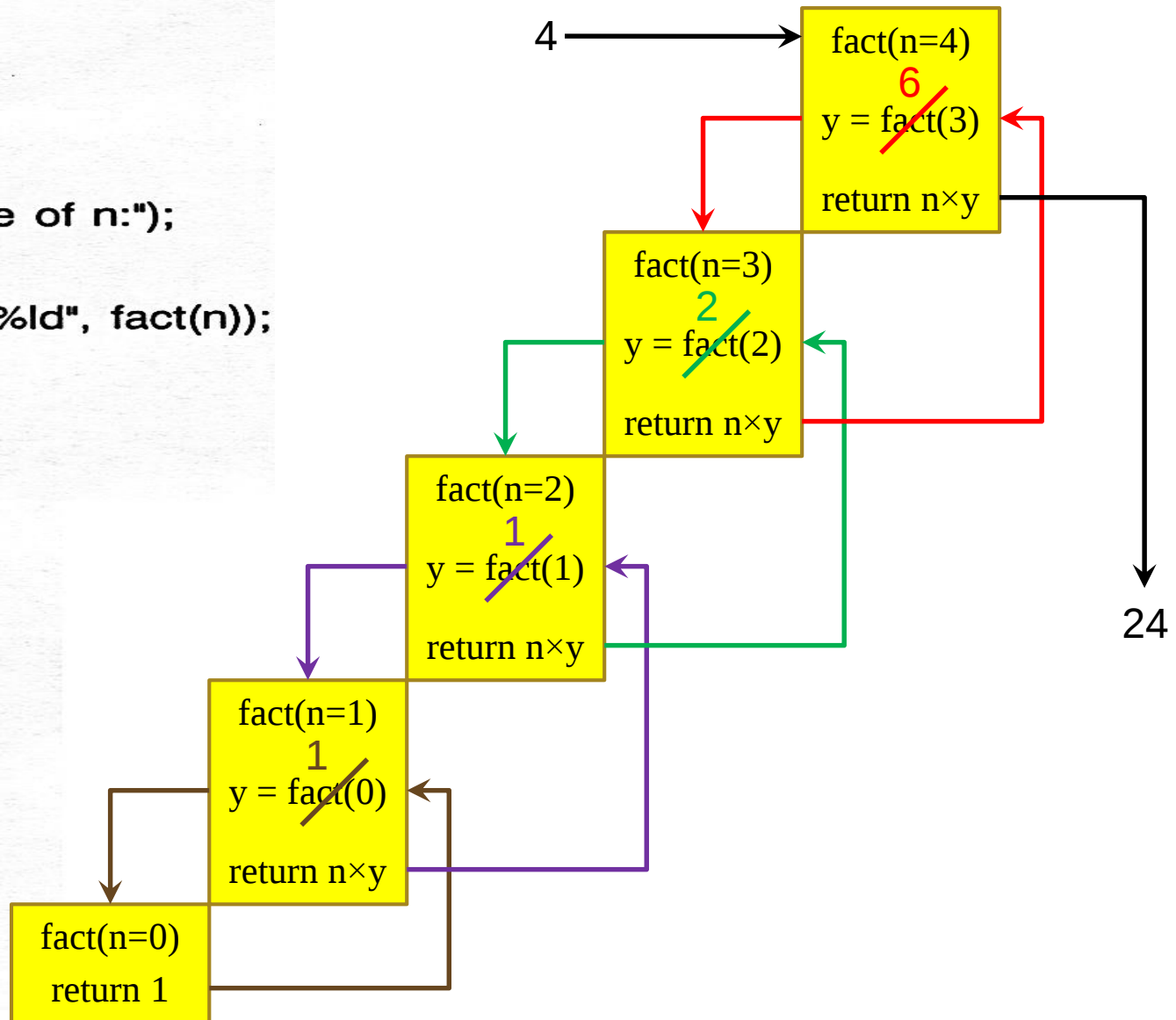
- به دلیل امکان فراخوانی تو در توی توابع و یا فراخوانی پشت سر هم توابع، بایستی ساختمان داده ای که رکوردهای فعالیت توابع را در خود نگهداری می کند خاصیت LIFO داشته باشد.
- پشته ساختمان داده ای با این ویژگی است و به دلیل اینکه در زمان اجرا دستکاری می شود، به آن **پشته زمان اجرا** می گویند.

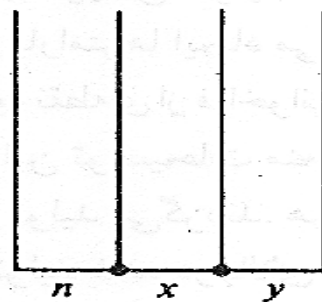
```

#include "stdio.h"
#include "conio.h"
long int fact(int);
int main()
{
    int n;
    clrscr();
    printf("Enter value of n:");
    scanf("%d", &n);
    printf("fact(n) = %ld", fact(n));
    getch();
    return 0;
}
//*****
long int fact(int n)
{
    int x, y;
    if(n == 0)
        return 1;
    x = n - 1;
    y = fact(x);
    return n * y;
}

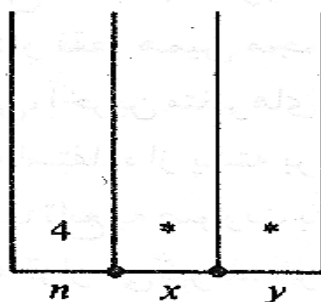
```

**مثال:** برنامه ای که توسط توابع بازگشتی، فاکتوریل یک عدد صحیح را محاسبه می کند.

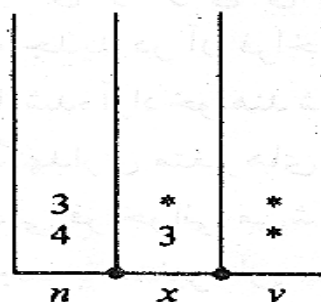




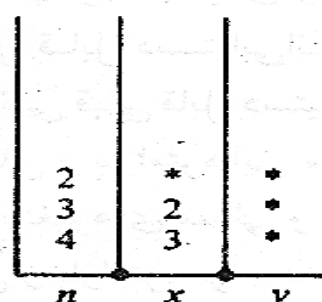
(الف) در آغاز



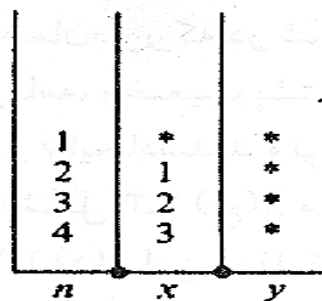
(ب)  $fact(4)$



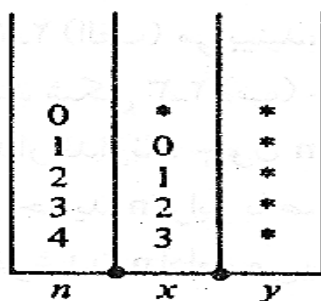
(ج)  $fact(3)$



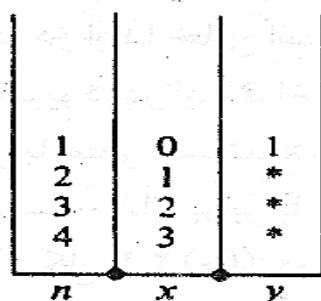
(د)  $fact(2)$



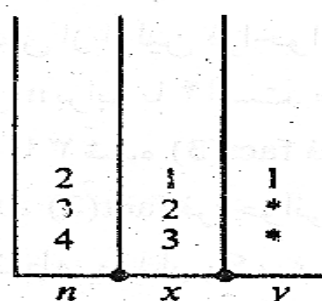
(ه)  $fact(1)$



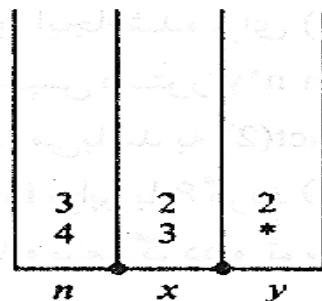
(و)  $fact(0)$



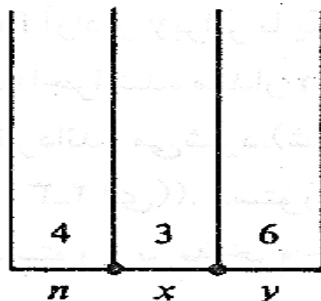
(ز)  $y = fact(0)$



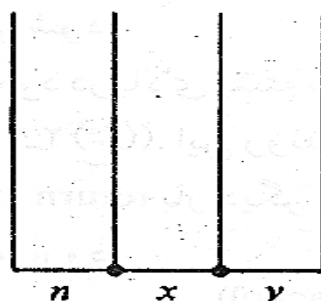
(ح)  $y = fact(1)$



(ط)  $y = fact(2)$



(ی)  $y = fact(3)$



(ک)  $printf(\%d, fact(4))$

محتویات پشته در فراخوانی  $fact(4)$ .

# ارزیابی عبارت ها

# کاربردهای پشته

- عبارت های عملیاتی به سه شکل نوشته می شوند :

- (۱) عبارت پیشوندی : عملگر **قبل** از عملوندها قرار می گیرد.  
 $+A B$
- (۲) عبارت میانوندی : عملگر **بین** عملوندها قرار می گیرد.  
 $A + B$
- (۳) عبارت پسوندی : عملگر **بعد** از عملوندها قرار می گیرد.  
 $A B +$

- اکثر کامپایلرها عبارت های میانوندی را به عبارت پسوندی تبدیل نموده و آنها را استفاده می کنند.

- برای تبدیل عبارت های میانوندی و پسوندی به یکدیگر از ساختمان داده پشته استفاده می شود.

# تبدیل عبارت میانوندی به پسوندی

**تذکر!** قبل از شرح عملیات تبدیل لازم است تذکر داده شود که عبارت ها به شکل میانوندی دارای اولویت می باشند، که در تبدیل ها باید مد نظر قرار گیرد.

| عبارت میانوند                            | عبارت پسوند                       |
|------------------------------------------|-----------------------------------|
| $A + B$                                  | $AB+$                             |
| $A + B - C$                              | $AB + C-$                         |
| $(A + B) * (C - D)$                      | $AB + CD -*$                      |
| $A \wedge B * C - D + E / F / (G + H)$   | $AB \wedge C * D - EF / GH + / +$ |
| $((A + B) * C - (D - E)) \wedge (F + G)$ | $AB + C * DE -- FG + \wedge$      |
| $A - B / (C * D \wedge E)$               | $ABCDE \wedge * / -$              |

## الگوریتم تبدیل عبارت میانوندی به پسوندی

دریافتی: عبارت میانوند

برگشتی: عبارت پسوندی

۱. پشته‌ای خالی برای عملگرها ایجاد کنید.

۲. تازمائی که خطایی رخ نداده است و به انتهای عبارت میانوند نرسیدی اعمال زیر را انجام بده:  
الف: نشانه بعدی (ثابت، متغیر، عملگر، پرانتز باز، پرانتز بسته) را از عبارت میانوند دریافت کن.  
ب: اگر نشانه

- پرانتز باز است، آن را در پشته قرار بده.

- 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 (اگر پشته خالی شد و به پرانتز باز نرسیدید، خطایی در عبارت میانوند وجود دارد).

- 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. این کار را برای عملگر جدید موجود در بالای پشته تکرار کنید.

توجه: تقدم پرانتز بازی که در پشته قرار دارد، از تقدم سایر عملگرها کمتر است.

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

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



تبدیل عبارت میانوند  $(A - (B + C)) / (E + F)$  به عبارت پسوندی. در این تبدیل دقت داشته باشید که تقدم پرانتز بازی که در خارج از پشته قرار دارد، از تقدم تمام عملگرها بیشتر است و تقدم پرانتز بازی که در داخل پشته قرار دارد، از تقدم تمام عملگرها کمتر است. این تبدیل به صورت جدول زیر است:

| پشته | عبارت پسوندی | کاراکتر |
|------|--------------|---------|
| (    | A            | (       |
| (    | A            | A       |
| (-   | A            | -       |
| (-(  | A            | (       |
| (-(  | AB           | B       |
| (-(+ | AB           | +       |
| (-(+ | ABC          | C       |
| (-   | ABC+         | )       |
|      | ABC+-        | )       |
| /    | ABC+-        | /       |
| /((  | ABC+-        | (       |
| /((  | ABC+-E       | E       |
| /((+ | ABC+-E       | +       |
| /((+ | ABC+-EF      | F       |
| /    | ABC+-EF+     | )       |
|      | ABC+-EF+/-   |         |

## الگوریتم ارزیابی عبارات پسوندی

دریافتی: عبارت پسوندی

برگشتی: مقدار عبارت پسوندی

۱. پشته‌ای خالی برای ذخیره عملوندها ایجاد کنید.

۲. اعمال زیر را انجام دهید تا به انتهای عبارت برسید:

۲-۱. نماد بعدی را از عبارت پسوندی دریافت کنید.

۲-۲. اگر نماد یک عملوند است آن را در پشته قرار دهید و اگر عملگر است به صورت زیر عمل کنید:

۲-۲. دو مقدار را از پشته خارج کنید.

۲-۴. عملگر را بر روی این دو مقدار اجرا کنید.

۲-۵. نتیجه را در پشته ذخیره کنید.

۳. وقتی به انتهای عبارت پسوندی رسیدی، مقدارش در بالای پشته قرار دارد.

## مثال ۷-۲

ارزیابی عبارت پسوندی - + 5 9 \* 4 2.

۲ در پشته قرار می‌گیرد.

۴ در پشته قرار می‌گیرد.

۴ و ۲ از پشته حذف می‌شوند، در هم ضرب شده،

نتیجه در پشته قرار می‌گیرد.

۹ در پشته قرار می‌گیرد.

۵ در پشته قرار می‌گیرد.

۵ و ۹ از پشته حذف شده، جمع می‌شوند و

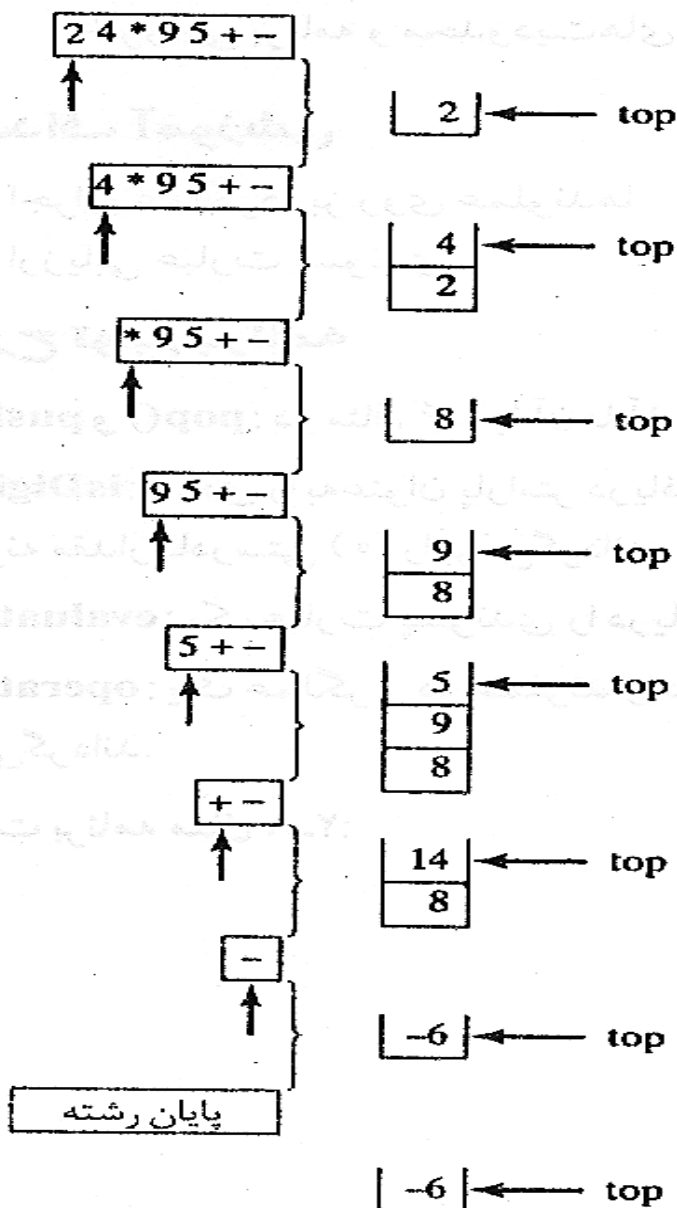
نتیجه در پشته قرار می‌گیرد.

۱۴ و ۸ از پشته حذف شده، از هم کم می‌شوند و

نتیجه در پشته قرار می‌گیرد.

مقدار عبارت در بالای پشته است.

پشته عبارت



# تبدیل عبارت میانوندی به پیشوندی

- الگوریتم تبدیل عبارت میانوندی به پیشوندی
- کنفرانس با دانشجویان

