

Stack یا پشته

Stack لیستی است که اعمال ورودی و خروجی یا اضافه و حذف در آن از یک طرف لیست انجام می‌شود. به این جهت به آن لیست Last In First Out (LIFO) می‌گویند. بدین معنی که آخرین ورودی به پشته، اولین خروجی خواهد بود. عنصر بالایی پشته را top پشته می‌گویند. با افزودن داده روی پشته، متغیر top یکی زیاد شده و داده در محل top از پشته قرار می‌گیرد. برای خارج کردن یک عنصر از پشته نیز داده‌ای که در محل top قرار گرفته از Stack خارج می‌گردد و متغیر top یکی کم می‌شود. مقدار اولیه top صفر است و با افزودن داده به یک پشته n عضوی، top می‌تواند تا مقدار n تغییر کند.

Top = 0 ==> پشته خالی است

Top = n ==> پشته پر است

دو عمل اصلی برای پشته‌ها را با push کردن و pop کردن می‌شناسیم. Push (x) داده x را در بالای پشته قرار می‌دهد و عمل pop عنصر بالای پشته را در متغیر x ذخیره می‌کند.

$x = \text{pop} \equiv \text{pop}(x)$

Stack : Array [1 .. n] of items

```

int pop ( )
{
    int x ;
    if (top = 0)
    {
        C out << " پشته خالی است " ;
        return - 1 ;
    }
    else
    {
        x = Stack [top] ;
        top = top - 1 ;
    }
    return x ;
}

void push (int x)
{
    if (top == n)
    {
        C out << " پشته پر است " ;
        return - 1 ;
    }
    else
    {
        top ++ ;
        Stack [top] ;
    }
}

```

مثال : مقدار نهایی A و B و C چقدر است؟

n = 5 A = 10 B = 2 C = 5

push (B)

push (A + B)

pop (C)

push (A - B)

push (C)

push (B)

pop (A)

pop (B)

push (A × B)

push (C)

push (A)

pop (B)

pop (C)

pop (A)

2	
2	12
12	24
12	8
2	

A	B	C
10	2	5
2	12	12
24	2	12

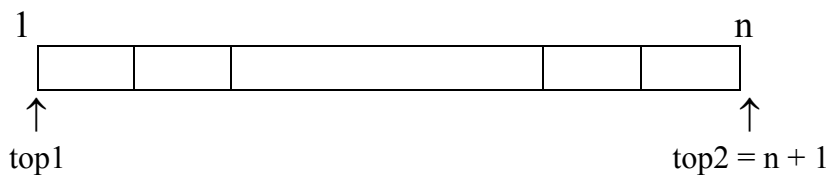
پشته‌های چندگانه

پشته دوگانه : برای پیداسازی دو پشته در یک آرایه نیاز به دو متغیر $top1$ برای نشان دادن بالاترین عنصر پشته اول و $top2$ برای بالاترین عنصر پشته دوم داریم. $top1$ و $top2$ در جهت عکس یکدیگر حرکت می‌کنند. مقدار اولیه $top1 = 0$ و مقدار اولیه $top2 = n + 1$ است.

$top1 = 0$ \Rightarrow پشته ۱ خالی است

$top2 = n + 1$ \Rightarrow پشته ۲ خالی است

$top2 = top1 + 1$ \Rightarrow آرایه پر است



دنباله‌های قابل قبول در پشته‌ها

هرگاه اعدادی را به صورت مرتب شده صعودی داشته باشیم و بخواهیم اعداد دیگری را از آن استخراج کنیم باید این قانون را رعایت کنیم که اعداد بزرگتر در صورتیکه اعداد کوچکتر در پشته قرار نگرفته‌اند حق قرار گرفتن در پشته را ندارند. مثلاً اعداد ۱, ۲, ۳, ۴ را در نظر می‌گیریم. عدد ۲ در صورتی می‌تواند push شود که حتماً عدد ۱ push شده باشد و عدد ۳ زمانی می‌تواند push شود که اعداد ۱ و ۲ قبلاً push شده باشند.

مثال : چهار عدد ۱, ۲, ۳, ۴ را داریم. کدامیک از اعداد زیر را می‌توانیم تولید کنیم؟

۲ ۱ ۳ ۴	۳ ۱ ۴ ۲	۳ ۲ ۴ ۱	۴ ۲ ۳ ۱	۴ ۳ ۱ ۲
push 1	push 1	push 1	push 1	push 1
push 2	push 2	push 2	push 2	push 2
pop 2	push 3	push 3	push 3	push 3
pop 1	pop 3	pop 3	push 4	push 4
push 3		pop 2	pop 4	pop 4
pop 3		push 4		pop 3
push 4	قابل تولید نیست	pop 4	قابل تولید نیست	قابل تولید نیست
pop 4		pop 1		

اگر اعداد به صورت صعودی داده شوند (۱ ۲ ۳ ۴) و سه عدد a و b و c داشته باشیم بطوریکه $b < c < a$ در اینصورت دنباله abc قابل تولید نیست.