

پشته کاغذی 1

این تمرین بری بالا رفتن سطح دانش شما از ساختمان داده طراحی شده است. در این دسته سوالات شما باید راه حل رو با ذکر مثال (در صورت نیاز) توضیح دهید. همچنین از ارائه پاسخ کوتاه خودداری کنید.

1) تفاوت بین استک و صف را توضیح دهید.

2) چگونه می‌توان یک استک را با استفاده از دو صف پیاده‌سازی کرد؟ مثال بزنید

3) استک‌ها چگونه در اجرای فراخوانی توابع بازگشتی استفاده می‌شوند؟

4) توضیح دهید که Underflow و Overflow چه زمانی اتفاق می‌افتد؟

5) توضیح دهید که چرا به postfix و prefix احتیاج داریم؟

پشته کاغذی 2

پرسش ۱: به بخش های مختلف سوال پاسخ دهید. در هر بخش پاسخ یکی از موارد را با رسم stack بنویسید.

الف (فرم prefix عبارت زیر را بنویسید.

1) $A - B / (C * D ^ E)$

2) $K + L - M * N + (O ^ P) * W / U / V * T + Q$

ب (فرم postfix عبارت زیر را بنویسید.

1) $A - B - D * E / F + B * C$

2) $(A + (B - C)) * ((D - E) / (F + G - H))$

پ (فرم infix عبارت زیر را بنویسید.

Postfix Expression: $A \ B \ C \ G \ + \ F \ + \ E \ * \ * \ *$

Prefix Expression: $+ \ * \ H \ * \ * \ * \ + \ A \ * \ + \ B \ C \ D \ F \ G \ E \ J$

پشته ی زنجیره ای

در این سوال شما باید قطعه کد مربوطه را بنویسید. شاید بپرسید چطوری؟ به طوری که یک پشته با خواص گفته شده نوشته شود و همچنین در آخر کد خودتون نمونه مثال برای خروجی نمایش بدین. یعنی یک استک شخصی سازی شده با ویژگی زیر پیاده سازی میکنید و براش یک نمونه ورودی تست میکنید.

سوال!

با استفاده از لیست پیوندی یک طرفه، پشته ای را پیاده سازی کنید. لازم به ذکر است عملیات Push , Pop میبایست در زمان $O(1)$ انجام شود.

پیاده پیاده تا پشته

منظور از getMin برگرداندن کوچکترین عنصر و getMax برگرداندن بزرگترین عنصر در ساختمان داده است . حال ساختمان داده پشته را طوری بنویسید که علاوه بر عمل push , pop عملیات های getMax , getMin را انجام دهد!!!

مثلا !

```
1  #include <iostream>
2  using namespace std;
3  class stack
4  {
5      public:
6          stack(int size) {}
7          void push(int x) {}
8          void pop() {}
9          int getMin() {}
10         void print() const {}
11         ~stack() {}
12
13     private:
14 };
15
16 int main()
17 {
18     stack ss(10);
19     ss.push(5);
20     ss.push(3);
21     ss.push(7);
22     ss.push(2);
23     ss.print();
24     cout << "Minimum: " << ss.getMin() << endl;
25     ss.pop();
26     cout << "Minimum after pop: " << ss.getMin() << endl;
27     ss.pop();
28     cout << "Minimum after pop: " << ss.getMin() << endl;
29 }
```

```
30 |     return 0;  
31 | }  
32 |
```

مرتب سازی استک

- محدودیت زمان: ۱ ثانیه
- محدودیت حافظه: ۲۵۶ مگابایت

با استفاده از یک stack کمکی، مقادیر داخل یک stack را مرتب کنید. (در این سوال حتما باید استک را خودتان پیاده سازی کنید).

مثال

ورودی نمونه ۱

40 20 63 42 77 84 13 43 15 109 11 1 38

خروجی نمونه ۱

1 11 13 15 20 38 40 42 43 63 77 84 109

بچرخونش

برنامه ای بنویسید که یک رشته به عنوان ورودی دریافت کرده و با استفاده از استک آن را معکوس کند. به عنوان مثال، اگر ورودی رشته "hello" باشد، خروجی باید "olleh" باشد.

نکات: برای این کار از عملیات‌های push و pop استفاده کنید.

ورودی نمونه ۱

hello

خروجی نمونه ۱

olleh

ورودی نمونه 2

HHIIi

ورودی نمونه ۱

iIIHH

فیلترینگ اعداد

الگوریتم بنویسید که در خط اول لیستی از اعداد را با فاصله از کاربر گرفته و سپس در خط دوم عدد صحیح K را از کاربر دریافت کند حال اعداد را در استک ذخیره نماید و سپس اعداد بزرگتر از K را به ترتیب نزولی چاپ کند (باید برای ذخیره اعداد از عملیات push و برداشت اعداد از عملیات pop استفاده شود)

نکات:

- از استک برای ذخیره اعداد بزرگتر از K استفاده کنید.
- باید توجه داشته باشید که مقادیر کوچکتر از K را نادیده بگیرید.

مثال‌ها:

ورودی نمونه ۱

4 5 22 8 9 7
8

خروجی نمونه ۱

22 9

توضیحات: لیست اعداد داده شد و کاربر عدد 8 رو در خط بعد به عنوان k وارد کرد حال اعداد 22 و 9 به ترتیب نزولی در خروجی چاپ شد چون تنها این دو عدد بزرگتر از عدد 8 بودن