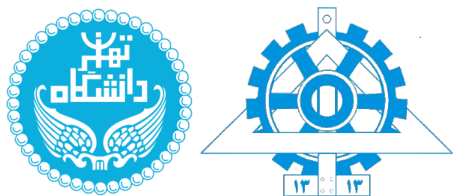


بہ نام خدا




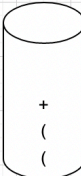
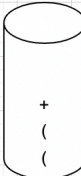
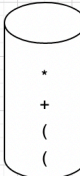
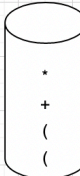


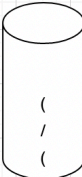
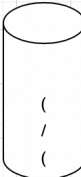
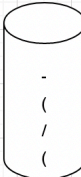
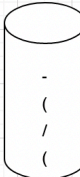
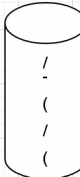
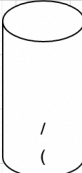






دانشگاه تهران، دانشکده مهندسی برق و کامپیوتر  
ساختمان های داده و الگوریتم

استاد: دکتر هشام فیلی  
کویز شماره ۲  
مدت زمان: ۴۰ دقیقه  
۱۴۰۰/۸/۹
















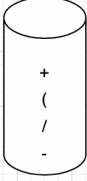

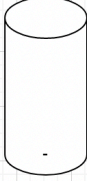



شماره دانشجویی خود را به صورت abcd1018 در نظر بگیرید و سوالات را طبق فرمول داده شده براساس شماره دانشجویی خود حل کنید. بارم هر سوال 5 نمره است.

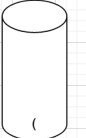
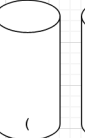

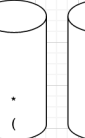
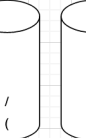
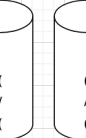

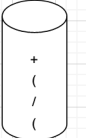
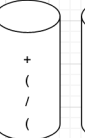

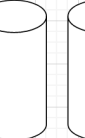
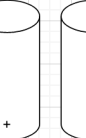
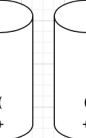

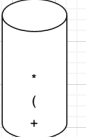
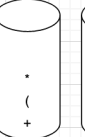
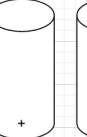
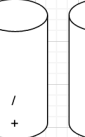
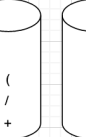
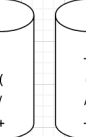

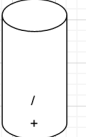
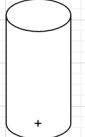

۱. با استفاده از استک عبارت infix زیر را به postfix تبدیل کرده و حالت استک را در هر مرحله بکشید. ابتدا باقی مانده عدد  $b+c$  بر ۳ را به دست آورید و سپس سوال متناظر را حل کنید.

						
		2	2	23	23	236
						
236*+	236*+	236*+	236*+9	236*+9	236*+98	236*+982
						
236*+982/-	236*+982/-/	236*+982/-/9	236*+982/-/9+	236*+982/-/9+	236*+982/-/9+7	236*+982/-/9+7*

.

						
	3	3	39	39*	39*2	39*2+
						
39*2+	39*2+4	39*2+4/	39*2+4/10	39*2+4/10	39*2+4/10	39*2+4/108
						
39*2+4/108	39*2+4/1081	39*2+4/1081+	39*2+4/1081+/-			

۲.

						
	8	8	89	89*	89*	89*2
						
89*2	89*26	89*26+	89*26+/-	89*26+/-	89*26+/-	89*26+/-4
						
89*26+4	89*26+412	89*26+412*	89*26+412*	89*26+412*	89*26+412*1	89*26+412*18
						
89*26+412*18+	89*26+412*18+/-	89*26+412*18+/-				

۲. باقی مانده d بر ۳ را به دست آورید و سوال متناظر را حل کنید

• ابتدا یک متغیر min برای ذخیره کردن مقدار مینیمم نگه میداریم. حال دو تابع push و pop را به اینصورت پیاده سازی میکنیم:  
 برای push اگر استک خالی باشد، عنصر جدید را در استک وارد میکنیم و min را برابر این عنصر میگذاریم. در غیر اینصورت باید مقدار عنصر جدید را با min مقایسه کنیم. اگر این عنصر بزرگ تر و یا مساوی min بود صرفاً آن را اضافه میکنیم در غیر اینصورت مقدار عنصر جدید را دو برابر کرده و از min کم میکنیم و سپس در استک قرار میدهیم و مقدار min را برابر عنصر جدید میگذاریم.  
 برای pop عنصر سر استک را خارج میکنیم. اگر این عنصر بزرگ تر و یا مساوی min بود، min تغییری نمیکند در غیر اینصورت min با مقدار (دو برابر min منهای مقدار عنصر pop شده) آپدیت میشود.

۱. برای این کار از دو آرایه استفاده میکنیم. در یک آرایه k تایی top هر استک را نگه میداریم و با استفاده از یک آرایه n تایی خانه ی بعدی هر خانه را مشخص میکنیم

۲. برای عمل enqueue هر بار تمام عناصر را از استک اول به ترتیب به استک دوم منتقل میکنیم و عنصر مورد نظر را در استک اول قرار میدهیم و سپس تمام عناصر را باز میگردانیم.  
 برای عمل dequeue عنصر اول را پاپ میکنیم.

۳. یک لینک لیست یک طرفه در اختیار داریم. باقی مانده c را بر دو به دست آورید و سوال متناظر را حل کنید

```

1 detectLoop(head):
2     fptr = head
3     sptr = head
4
5     while fptr is not nil and sptr is not nil
6         and sptr->next is not nil:
7
8         sptr = sptr->next->next
9         fptr = fptr->next
10
11        if fptr is equal to sptr:
12            return true
13
14    return false
15

```

۱.

```

1  mergeLists(head1, head2):
2      dummyNode = new Node
3      tail = dummyNode
4      dummyNode->next = nil
5      while true:
6          if head1 is equal to nil:
7              tail->next = head2
8              break
9          else if head2 is equal to nil:
10             tail->next = head1
11             break
12
13         if head1->data is less than or equal to
14             head2-> data:
15
16             temp = head1
17             head1 = temp->next
18             temp->next = tail->next
19             tail->next = temp
20         else:
21             temp = head2
22             head2 = temp->next
23             temp->next = tail-> next
24             tail->next = temp
25
26         tail = tail->next
27     return dummyNode->next

```

۴. صفی شامل  $n$  فرد با قدهای متمایز داریم. ۲ نفر میتوانند همدیگر را ببینند، در صورتی که تمام افراد بین این دو قدشان از هر دوی این دو نفر بیشتر نباشد. الگوریتمی برای پیدا کردن تعداد جفت افرادی که میتوانند همدیگر را ببینند پیدا کنید

با استفاده از داده ساختار استک مساله را حل میکنیم. از اول صف شروع میکنیم و هر مرحله قد افراد و تعداد افرادی را که میتوانند ببینند (count) (که در ابتدا صفر است) روی استک میگذاریم. اگر فرد جدید قد بیشتری از تاپ استک داشت همه افراد را pop میکنیم و در این فرایند مقدار count فریم پاپ شده را با count تاپ استک جمع میکنیم. این کار را تا خالی شدن استک ادامه میدهیم و در آخر فرد جدید را push میکنیم. برای هر شخص مقدار count در هنگام pop تعداد افرادیست که میتواند ببیند