1- خیر مشکل کد این است که اگر تعداد پرانتز های ')' بیشتر باشد آنگاه با وجود Pop و Push باز هم عبارت balanced چاپ می‌شود ← شکل صحیح کد رو ← زبان C++ نوشتم

2- کد ← زبان C++ نوشته شده

3- کد ← زبان C++ نوشته شده

4- تعداد کل حالات برای رسیدن ← ترتیب [3,2,1] 6 = 3! می‌باشد. اما ما باید حالات مطلوب را بیابیم. حالت 1: 3,2,1 ← از ابتدا ← ترتیب است و نیازی ← عملیات اضافه ندارد. حالت 2: 2,3,1 ← عدد 2 و 3 را مستقل و 1 را آنقدر می‌دهیم سپس آنها را برگردانیم

حالت 3: 3,1,2 ← عدد 3 را وارد صف کرده و اعداد 1 و 2 را جابه جا می‌کنیم.

حالت 4: 1,3,2 ← عدد یک ← صف رفته و 2 و 3 نیز باقی مانده و صدیقا می‌شوند.

حالت 5: 2,1,3 ← 2 وارد صف شده، 1 را آنقدر می‌دهیم و 3 را جابه جا می‌کنیم.

حالت 6: 1,2,3 ← این حالت امکان پذیر نیست.

5- در ابتدا پشته خالی است و همچنین می‌دانیم که نمی‌توانیم بیشتر از عناصر موجود در پشته، عنصر حذف کنیم. آنرا فرض کنیم $n = ck$ باشد داریم: عملیات پشته $k+1 \rightarrow O(2k)$

$c(k+1) \rightarrow O(2ck) \approx O(2n) \rightarrow \underline{O(n)}$

Year.....Month.....Day..... Subject:.....

6- در یک عبارت پسوندی حداکثر فضای مورد نیاز پشته برای ارزیابی یک عبارت پسوندی برابر است با بیشترین تعداد عملوندهای هم زمان موجود در پشته.

7- مراحل: % عملوند است پس به خروجی اضافه می‌شود / - عملگر است ← پشته اضافه می‌شود / y عملوند است ← خروجی اضافه می‌شود xy* / - عملگر است پس وارد پشته شده و - قبلی خارج می‌شود xy* - / z عملوند است به خروجی اضافه می‌شود xy*z- / * عملگر و اولویت آن بیشتر از - است پس به پشته اضافه می‌شود (*-) / ( ) عملگر و اولویت آن از همه بیشتر است پس به پشته اضافه می‌شود. (*-( / a عملوند است و به خروجی اضافه می‌شود xy*za- / + عملگر است و چون بعد از پرانتز آمده به پشته اضافه می‌شود (+(*- / b عملوند است به خروجی اضافه می‌شود xy*zab- / ) پرانتز بسته است پس همهٔ عملگرها را تا رسیدن به پرانتز باز وارد خروجی می‌کنیم و پرانتز را حذف می‌کنیم. xy*zab+- / در نهایت کل پشته را وارد خروجی می‌کنیم. **xy*zab+*-**

10- کد به زبان C++ نوشته شده

11- کد به زبان C++ نوشته شده

12- کد به زبان C++ نوشته شده