Tam Nguyen

Cs120 \_ LAB 8

**Progamming Excercise 3:**

|  |  |  |  |
| --- | --- | --- | --- |
| Test Plan 6-4 | | | |
| Test Case | Commands | Expected result | Checked |
| Valid expression with parentheses | 3\*(a+b) | Valid | X |
| Valid expression with mixed delimiters | F[3\*(a+b)] | Valid | X |
| Invalid expression with mixed delimiters | (f[b) - (c+d]/2 | Not valid | X |
| Empty experession | Empty string <newline> | True | X |
| Improperly paired brace | A = f[+b | False | X |

**Analysis Exercise 1:**

Part A:

|  |  |
| --- | --- |
| abc > abc  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch; | abc > bac  cin >> ch; permuteStack.push(ch);  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  ch = permutestack.pop(); cout << ch;  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch; |
| abc > cab  We can't because pop a always after pop b | abc > acb  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  cin >> ch; permuteStack.push(ch);  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  ch = permutestack.pop(); cout << ch; |
| abc > bca  cin >> ch; permuteStack.push(ch);  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  ch = permutestack.pop(); cout << ch; | abc > cba  cin >> ch; permuteStack.push(ch);  cin >> ch; permuteStack.push(ch);  cin >> ch; permuteStack.push(ch);  ch = permutestack.pop(); cout << ch;  ch = permutestack.pop(); cout << ch;  ch = permutestack.pop(); cout << ch; |

**Part B:**

The only permutation can perform is dcba.

If we want to have d as the beginning of output string, then we need to push abc before push d. Hence, we can only pop the top character in the stack consequently, which is d-c-b-a.