CSN-261: Data Structures Laboratory

Lab Assignment 4 (L4)

Instructions:

1. **Use either C/C++ for solving the assignment.**
2. **Array index starts with 0 in C/C++.**
3. **RED color indicates the input in each test case.**

Problem 1: With an ordered balanced **Square Bracket, i.e., []/ Curly Brackets, i.e., {}/ Bracket, i.e., ()**, the problem is to recognise the input sentences composed of sequences of these symbols like [,], {, }, (and), which are correctly nested following the priority order, i.e., **[]>{}>()**. **Following this order, write a program to determine the type of error (if exists) or its correctness in the input sentence having the above mentioned symbols. Limit of the number of such symbols in a sentence is 20. The error types can be: Unbalancing of LHS/RHS [, {, (or], },) and Missing Priority Order of []>{}>().**

Correct examples:

[( )]
{ }
[{( )}]
[{{{( )}}}]
[[[{ }]]]
(( ))

Wrong examples:

[} *Error: Unbalanced [and }*
(} *Error: Unbalanced (and }*
(( ) *Error: Unbalanced)*
{[ ]} *Error: Missing Priority of { } and [ ]*
([ ]) *Error: Missing Priority of ( ) and [ ]*
[{( )}] *Unbalanced)*

All these inputs can be considered as the test cases.

Problem 2: Consider **three polynomial expressions** stored in three different **singly linked lists**. Write a program to add these polynomial expressions and print the final linked list.

Test Case:

Input:

1st number = 5x^2 + 4x^1 + 2x^0

2nd number = 5x^1 + 5x^0

3rd number = 7x^5 + 5x^3+6x^0

Output:

7x^5+5x^3+5x^2 + 9x^1 + 13x^0

Problem 3: Consider an ordered linked list of army soldiers **(*S1->S2->S3…->Sn*),** where ***S1, S2, S3, …, Sn*** are the soldier **IDs** in **SORTED** order. They need to **GO** from the ground floor to the top floor of a building following the same **SORTED** order. To do this, they use ***m*** lifts. The capacity of each lift is ***c = n/m***, where ***n*** and ***m*** are chosen such that ***c*** should be an integer value (e.g., ***n = 20, m = 4,*** then ***c = 5***). Thus, this list is divided into ***m*** slices (each of size ***c***) obeying the same **SORTED** order of soldiers IDs. Each slice of soldiers get **ENTRY** into a **SEPARATE** lift following the same **SORTED** order (at the ground floor) and **EXIT** in **LIFO** order on the top floor. At a time, only one soldier exits from the **RANDOM** lift following the **LIFO** order and rebuilds the **SAME SORTED** list. Write a program to implement the above mentioned problem.

Test Case:

Print: Enter the values for *n, m* and *c*: 20 4 5 (***n = 20, m = 4*** then ***c = 5***)

Print: Soldiers in the respective Lifts at the Ground Floor are as:

Print: Soldiers in Lift 1: *S1, S2, S3, S4, S5*

Print: Soldiers in Lift 2: *S6, S7, S8, S9, S10*

Print: Soldiers in Lift 3: *S11, S12, S13, S14, S15*

Print: Soldiers in Lift 4: *S16, S17, S16, S19, S20*

Soldiers exit form the random lifts at the top floor are as:

Print: RANDOM SOLDIER ID (e.g., S5) popped from the respective Lift (Lift 1)

Print: Final SORTED LIST is: S5

Print: RANDOM SOLDIER is (e.g., S15) popped from the respective Lift (Lift 3)

Print: Final SORTED LIST is: S5 S15

Print: RANDOM SOLDIER ID (e.g., S4) popped from the respective Lift (Lift 1)

Print: Final SORTED LIST is: S4 S5 S15

Print: RANDOM SOLDIER ID (e.g., S4) popped from the respective Lift (Lift 4)

Print: Final SORTED LIST is: S4 S5 S15 S20

<DO IT for 20 times for each soldier……..>

Print: The final SORTED is: *S1, S2, …, S20*.