Exercises

Review Questions

1. What do you understand by stack overflow and underflow?
2. Differentiate between an array and a stack.
3. How does a stack implemented using a linked list differ from a stack implemented using an array?
4. Differentiate between `peek()` and `pop()` functions.
5. Why are parentheses not required in postfix/prefix expressions?
6. Explain how stacks are used in a non-recursive program?
7. What do you understand by a multiple stack? How is it useful?
8. Explain the terms infix expression, prefix expression, and postfix expression. Convert the following infix expressions to their postfix equivalents:
   (a) A – B + C (b) A * B + C / D
   (c) (A – B) + C * D / E – C
   (d) (A * B) + (C / D) – (D + E)
   (e) ((A – B) + D / ((E + F) * G))
   (f) (A – 2 * (B + C) / D * E) + F
   (g) 14 / 7 * 3 – 4 + 9 / 2
9. Convert the following infix expressions to their postfix equivalents:
   (a) A – B + C (b) A * B + C / D
   (c) (A – B) + C * D / E – C
   (d) (A * B) + (C / D) – (D + E)
   (e) ((A – B) + D / ((E + F) * G))
   (f) (A – 2 * (B + C) / D * E) + F
   (g) 14 / 7 * 3 – 4 + 9 / 2
10. Find the infix equivalents of the following postfix equivalents:

(a) A B + C * D – (b) ABC * + D –

11. Give the infix expression of the following prefix expressions.
(a) * – + A B C D (b) + – a * B C D

12. Convert the expression given below into its corresponding postfix expression and then evaluate it. Also write a program to evaluate a postfix expression.
10 + ((7 – 5) + 10)/2

13. Write a function that accepts two stacks. Copy the contents of first stack in the second stack. Note that the order of elements must be preserved.
(*Hint: use a temporary stack*)

14. Draw the stack structure in each case when the following operations are performed on an empty stack.
(a) Add A, B, C, D, E, F (b) Delete two letters
(c) Add G (d) Add H
(e) Delete four letters (f) Add I

15. Differentiate between an iterative function and a recursive function. Which one will you prefer to use and in what circumstances?

16. Explain the Tower of Hanoi problem.

Programming Exercises

1. Write a program to implement a stack using a linked list.

2. Write a program to convert the expression "a+b" into "ab+".

3. Write a program to convert the expression "a+b" into "+ab".

4. Write a program to implement a stack that stores names of students in the class.

5. Write a program to input two stacks and compare their contents.

Stack using Array: Assignments

Assignments:

1. Write a program to insert an element into the stack using an array (Push Operation).
2. Write a program to delete an element from the stack using an array (Pop Operation).
3. Write a program to return the value of the topmost element of the stack (without deleting it from the stack) using an array (Peep operation).
4. Write a program to display the elements of a stack using an array.

Stack using Linked List: Assignments

Assignments:

1. Write a program to insert an element into the stack using linked list (Push Operation).
2. Write a program to delete an element from the stack using linked list (Pop Operation).
3. Write a program to return the value of the topmost element of the stack (without deleting it from the stack) using linked list (Peep operation).
4. Write a program to display the elements of a stack using linked list .

Arithmetic Expressions: Assignments

Assignments (to be implemented using stack):

1. Write a program to reverse a list of given numbers.
2. Write a program to convert an infix expression into its equivalent postfix notation.
3. Write a program to evaluate a postfix expression.
4. Write a program to convert an infix expression to a prefix expression.
5. Write a program to evaluate a prefix expression.
6. Convert the following infix expression into postfix expression using appropriate algorithm showing all steps: A – (B / C + (D % E * F) / G)* H
7. Consider the infix expression: 9 – ((3 * 4) + 8) / 4. Convert it to the equivalent postfix expression first. Then evaluate the postfix expression using appropriate algorithm showing all steps.
8. Consider the infix expression: 9 – ((3 * 4) + 8) / 4. Convert it to the equivalent prefix expression first. Then evaluate the prefix expression using appropriate algorithm showing all steps.
9. Convert the following infix expression into prefix expression using appropriate algorithm showing all steps: (A – B / C) * (A / K – L).

Recursion: Assignments

Assignments (using recursion):

1. Write a program to calculate the factorial of a given number.
2. Write a program to calculate the GCD of two numbers using recursive functions.
3. Write a program to calculate exp(x,y) using recursive functions.
4. Write a program to print the Fibonacci series using recursion.
5. Write a program to solve the tower of Hanoi problem using recursion.