Ex No: 5 Implementation of Stack ADT

A stack is a last in, first out (LIFO) abstract data type and data structure. A stack can have any abstract data type as an element, but is characterized by only two fundamental operations: push and pop. The push operation adds to the top of the list, hiding any items already on the stack, or initializing the stack if it is empty. The pop operation removes an item from the top of the list, and returns this value to the caller. A pop either reveals previously concealed items, or results in an empty list.

The two applications of stack are expression evaluation and syntax parsing. Calculators employing reverse Polish notation use a stack structure to hold values. Expressions can be represented in prefix, postfix or infix notations. Conversion from one form of the expression to another form may be accomplished using a stack. Many compilers use a stack for parsing the syntax of expressions, program blocks etc. before translating into low level code. Most of the programming languages are context-free languages allowing them to be parsed with stack based machines.

Stack operations are:
Push() – inserts an item into stack
Pop() - removes an item from stack
Peep() – returns the top element of stack
isEmptyStack() – checks whether stack is empty
isFullStack() – checks whether stack is full

```
PUSH
if(top==max-1)
    print(stack overflow);
else
    top=top+1;
    stack arr[top]=pushed item;
    end if;
end algorithm;
```

```
POP
if (top==-1)
  print(stack underflow);
else
  print(popped element is stack arr[top]);
  top=top-1;
  end if;
end algorithm;
```

Two way stack

Modify Stack ADT so that two ends are used for inserting/deleting the elements in the same stack. Use suitable overflow/underflow condition when boundary for two stacks is fixed/not fixed.

For ex: Stack can hold maximum N elements

Stack 1 contains elements that are pushed from one end and Stack 2 contains elements that are pushed from the other end.

Stack 1 can have elements in the position from 0 to N/2

Stack 2 can have elements in the position from N-1 to (N/2)+1

Else Two way stack can also be created when N/2 is not fixed i.e varying number of elements for both the stacks.

Aim: To determine whether the given string is palindrome or not

Procedure:

Input: String

Output: Palindrome or not

begin

1.read input (str) from the user

2.loop

a. push(ch) each character into the stack

end loop

3.loop (stack not empty)

a. ch = pop()

b. Compare ch and str[index]

c. Print palindrome or not palindrome

end loop

end

Sample Questions, but not limited to

1. Reversing the string
2. Converting decimal number to binary number
3. Check whether the given string is palindrome or not
4. Implement Stack using Linked List AD
5. Implementation of Two way Stack
6. MoveStack – move stack S1 contents into another stack S2 without changing the order
7. Write a program to implement the Algorithm “reverse the number series”. Test your program with the number series 1, 3, 5, 7, 9, 2, 4, 6, 8

8. Write pseudo code for an algorithm that prints a decimal number as an octal or hexa decimal number
9. MergeStack – create new stack S3 from the contents of stack S1 and stack S2. Stack S3 contains S1 items first and then S2 items in the original order.
10. Split Stack - split stack S which contains integers. Split S into two stacks S1 and S2 such that S1 contains only positive numbers and S2 contains only negative numbers.
11. Write procedure minStack() such that it returns the minimum number of stack items.
12. Expression Evaluation (i) Balancing Parenthesis checking
    (i) ((a+ b) * c) (ii) (a * b) * c) (iii) A * (B + C))
13. Write Program to check the C source file created is with right number of { }.
14. Evaluate the following infix to postfix expression when A=1, B=2, C=3, D=4, E=5, and F=6.
    (i) D – B + C (ii) A * B + C * D
    (iii) (A + B) * C – D * F + C (iv) (A – 2 * (B+C) – D * E) * F
15. Determine the value of the following postfix expression when A is 2, B is 3, C is 4 and D is 5.
    (i) A B C D * - + (ii) D C * B A + -

Set A – 1 to 5
Set B – 6 to 11
Set C – 12 to 15

Assessment Rubrics for Ex 5:

Parameters	**Max Marks**	**Actual Marks**
Uniqueness of the problem selection	10	
Stack ADT procedure and Diagram Representation	10	
Solution for 3 different problems – set A , B, C	10	
Viva (Online Test)	10	
Adherence to the template for documentation (Record)	10	
Total	50	
Staff Signature		