Postfix Evaluation

Algorithm:

Let P be an expression written in postfix expression, **Vstack** be an operand stack.

1. Begin.
2. Vstack = empty stack
3. Repeat steps 4, 5 and 6 until end of string, P.
4. Scan one char at a time from left to right in P.
5. If scan char is operand, then read its corresponding value and push it into vstack.
6. If scan char is operator, then,
   1. Pop and place into operand2.
   2. Pop and place into operand1.
   3. Compute : result = operand1 operator operand2
   4. Push result into vstack.
7. Pop vstack and display, which is required value of given expression.
8. End.

**Tracing:**

Evaluate the postfix expression:

ABC+\*CBA-+\* where A=1, B=2, C =3.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| S.N | Scan char | Value | Op1 | Op2 | Result | Vstack |
| 1 | A | 1 |  |  |  | 1 |
| 2 | B | 2 |  |  |  | 1, 2 |
| 3 | C | 3 |  |  |  | 1,2,3 |
| 4 | + |  | 2 | 3 | 5 | 1,5 |
| 5 | \* |  | 1 | 5 | 5 | 5 |
| 6 | C | 3 |  |  |  | 5,3 |
| 7 | B | 2 |  |  |  | 5,3,2 |
| 8 | A | 1 |  |  |  | 5,3,2,1 |
| 9 | - |  | 2 | 1 | 1 | 5,3,1 |
| 10 | + |  | 3 | 1 | 4 | 5,4 |
| 11 | \* |  | 5 | 4 | 20 | 20 |
| Final Result of Expression : 20 | | | | | | |

**Exercise**

Evaluate the postfix expression:

ABC+-D\*EF+$ where, A=1, B=2, C=3, D=5, E=7, F=8