❖ Infix to Postfix:

- Explanation- The problem is to convert an infix expression (where operators are placed between operands) into a postfix expression (where operators follow their operands). This conversion is essential for evaluating expressions without the need for parentheses, making it easier for computers to process. The program should read an infix expression from the user and output the equivalent postfix notation.
- Time Complexity- O(n)
- Space Complexity- O(n)
- Flowchart-

```
    +-------------------+
|       Start       |
+-------------------+
         |
         v
+-------------------+
|   Input Expression |
+-------------------+
         |
         v
+-------------------+
| Initialize Stack   |
+-------------------+
         |
         v
+-------------------+
|   Read Token      |
+-------------------+
         |
         v
+-------------------+
|   Is Token a      |
|   Digit/Operand?  |
+-------------------+
      /   \
    Yes     No
     |       |
     v       |
+-------------------+
| Append Token to    |
| Output (Postfix)   |
+-------------------+
         |
         v
+-------------------+
|   Is Token '(' ?  |
+-------------------+
      /   \
    Yes     No
     |       |
     v       v
+-------------------+
| Push '(' to Stack |
+-------------------+
         |
         v
+-------------------+
|   Is Token ')' ?  |
+-------------------+
      /   \
    Yes     No
     |       |
     v       |
+-------------------+
| Pop from Stack    |
| to Output until   |
| '(' is found      |
+-------------------+
         |
         v
+-------------------+
|   Is Token an     |
|   Operator?       |
+-------------------+
      /   \
    Yes     No
     |       |
     v       |
+-------------------+
| Pop from Stack    |
| to Output while   |
| Precedence is ≥   |
+-------------------+
         |
         v
+-------------------+
| Push Operator to   |
| Stack              |
+-------------------+
         |
         v
+-------------------+
|   Is Stack Empty? |
+-------------------+
      /   \
    Yes     No
     |       |
     v       |
+-------------------+
|  Pop Remaining    |
|  Operators to Output |
+-------------------+
         |
         v
+-------------------+
|   Return Postfix  |
+-------------------+
         |
         v
+-------------------+
|       End         |
+-------------------+
```