Here are 5 challenging examples of **infix to postfix conversion** problems, along with step-by-step explanations and solutions:

---

## Example 1:

**Infix Expression**:
`A + B * (C ^ D - E) ^ (F + G * H) - I`

**Solution Steps:**

1. Start scanning the expression left to right:
2. Use a stack for operators and append operands to the postfix output.

| Step | Input Symbol | Stack (S) | Postfix (P) |
|---|---|---|---|
| 1 | A | (unchanged) | A |
| 2 | + | + | A |
| 3 | B | + | A B |
| 4 | * | + * | A B |
| 5 | ( | + * ( | A B |
| 6 | C | + * ( | A B C |
| 7 | ^ | + * ( ^ | A B C |
| 8 | D | + * ( ^ | A B C D |
| 9 | - | + * ( - | A B C D ^ |
| 10 | E | + * ( - | A B C D ^ E |
| 11 | ) | + * | A B C D ^ E - |
| 12 | ^ | + ^ | A B C D ^ E - |
| 13 | ( | + ^ ( | A B C D ^ E - |
| 14 | F | + ^ ( | A B C D ^ E - F |
| 15 | + | + ^ ( + | A B C D ^ E - F |
| 16 | G | + ^ ( + | A B C D ^ E - F G |
| 17 | * | + ^ ( + * | A B C D ^ E - F G |
| 18 | H | + ^ ( + * | A B C D ^ E - F G H |
| 19 | ) | + ^ | A B C D ^ E - F G H * + |
| 20 | - | - | A B C D ^ E - F G H * + ^ |
| 21 | I | - | A B C D ^ E - F G H * + ^ I |

**Postfix Expression:**

`A B C D ^ E - F G H * + ^ * + I -`

---

## Example 2:

**Infix Expression**:
`(A + B) * (C + D) / E`

**Postfix:**

```
A B + C D + * E /
```

---

## Example 3:

**Infix Expression**:

```
A * B + C / D ^ E - F
```

**Postfix:**

```
A B * C D E ^ / + F -
```

---

## Example 4:

**Infix Expression**:

```
(A + B * C) / (D - E ^ F)
```

**Postfix:**

```
A B C * + D E F ^ - /
```

---

## Example 5:

**Infix Expression**:

```
A ^ B ^ C + D * E - F / G
```

**Postfix:**

```
A B C ^ ^ D E * + F G / -
```

---

## Step-by-Step Explanation of Example 5:

1. **Precedence**: Exponentiation (^) has the highest precedence and is right-associative.
2. **Break Down**:
    - First handle `B ^ C`.
    - Then handle `A ^ (B ^ C)`.
    - Then evaluate `D * E` and `F / G`.
    - Combine terms with `+` and `-`.

---

These examples challenge precedence handling, associativity, and proper stack usage for grouping and nested expressions!

4o