Postfix Expression Evaluation - Detailed Breakdown

Introduction

Yeh document Postfix Expression Evaluation ke code ko detail mein samjhata hai. Iska pura breakdown, dry run, aur short notes diye gaye hain taaki aapko pura concept samajh aaye.

Code Snippet

```
#include<iostream>
#include<string>
#include<stack>
using namespace std;
int priority(char ch){
  if(ch=='+'||ch=='-') return 1;
  else if(ch=='*'||ch=='/')return 2;
}
int eval(int v1,int v2,char ch){
  if(ch=='+') return v1+v2;
  else if(ch == '-')return v1-v2;
  else if(ch == '*') return v1*v2;
  else return v1/v2;
}
int main(){
  string s = "126+4*8/+3-";
  stack<int> val;
  for(int i=0;i<s.length();i++){
    if(s[i]>=48 && s[i]<=57){
      val.push(s[i]-48);
    }
    else{
        int v2 = val.top();
        val.pop();
        char ch = s[i];
        int v1 = val.top();
        val.pop();
        int ans = eval(v1,v2,ch);
        val.push(ans);
        }
    }
    cout<<val.top();
```

Postfix Expression Evaluation - Detailed Breakdown

```
    return 0;
}
```

Thought Process Behind the Code

1. **Operands Handling:** Agar character ek digit hai, toh use integer me convert karke stack me push karo.
2. **Operators Handling:** Agar operator aaye toh stack se values nikal kar evaluation karo.
3. **Evaluation Logic:** Har operator ka corresponding operation perform karo aur answer wapas stack me push karo.
4. **Final Output:** Jab sab kuch process ho jaye, stack ke top par final answer milega.

Dry Run (Step-by-Step Execution)

Input: 126+4*8/+3-

Step	Character	Stack State
1	1	[1]
2	2	[1, 2]
3	6	[1, 2, 6]
4	+	[1, 8] (2+6=8)
5	4	[1, 8, 4]
6	*	[1, 32] (8*4=32)
7	8	[1, 32, 8]
8	/	[1, 4] (32/8=4)
9	+	[5] (1+4=5)
10	3	[5, 3]
11	-	[2] (5-3=2)

Final Output: 2

Postfix Expression Evaluation - Detailed Breakdown

Short Notes (Quick Revision)

- **Operands ko stack me push karo.**
- **Agar operator mile, toh stack se values nikal ke evaluation karo.**
- **Operator precedence ko dhyan me rakho.**
- **Jo last value stack me bachegi, wahi final answer hoga.**