

Đặt ngoặc

- Đầu vào là một phương trình theo kiểu string.
- Đầu ra là kết quả nhỏ nhất mà phương trình có thể đạt được.

```
int main () {  
    string exp = "3+4x5+6";  
    cout << MinValueOfExpression(exp) << endl;  
}
```

Đầu tiên ta cần một hàm kiểm tra xem có phải là phép toán hay không:

```
bool isOp(char c) {  
    return (c == '+' || c == 'x');  
}
```

Sau đó ta sử dụng hàm isOp để chia string exp thành 2 mảng là mảng numbers lưu các số và mảng ops lưu các phép toán:

```
int MinValueOfExpression (string exp) {  
    vector<int> numbers;  
    vector<int> ops;  
    string tmp;  
  
    for (int i = 0; i < exp.length(); i++) {  
        if (isOp(exp[i])) {  
            ops.push_back(exp[i]);  
            numbers.push_back(atoi(tmp.c_str()));  
            tmp = "";  
        }  
        else {  
            tmp += exp[i];  
        }  
    }  
  
    numbers.push_back(atoi(tmp.c_str()));  
}
```

```

5 int len = numbers.size();
6 vector<vector<int> > minValues;
7 minValues.assign(len, vector<int> (len, INT_MAX));
8
9 for (int i = 0; i < len; ++i)
10     minValues[i][i] = numbers[i];
11
12 for (int L = 2; L <= len; ++L) {
13     for (int i = 0; i < len - L + 1; ++i) {
14         int j = i + L - 1;
15         for (int k = i; k < j; ++k) {
16             if (ops[k] == '+')
17                 minValues[i][j] = min(minValues[i][k] + minValues[k + 1][j], minValues[i][j]);
18             else
19                 minValues[i][j] = min(minValues[i][k] * minValues[k + 1][j], minValues[i][j]);
20         }
21     }
22 }
23
24 return minValues[0][len-1];
25 }

```