Matrix Multiplication Assignment (Dynamic Programming)
Due: December 5, 2016

Given n matrices and their dimensions, write a function to determine the associative order of multiplication that minimizes the number of numeric multiplications required. As we know, if two matrices A and B have dimensions $r_A \times c_A$ and $r_B \times c_B$, multiplication is possible iff $c_A = r_B$. If this condition holds, the cost (number of numeric multiplications) is $r_A \times r_B \times c_B$ (or equivalently, $r_A \times c_A \times c_B$). Then, for example, if we have four matrices A, B, C and D, having dimensions 2×10, 10×5, 5×1 and 1×7, respectively, computing $(AB)(CD)$ requires 205 multiplications, whereas computing $(A(BC))D$ requires only 84 multiplications. Since returning merely a number gives us no clue as to how to achieve it, your function will return the associative order of multiplication as a string containing only asterisks and parentheses, with the asterisks as placeholders for the matrices, and the parentheses indicating the multiplicative grouping. For example, "$(*(**))*$" (quotes not included!) would indicate $(A_0(A_1A_2))A_3$. Specifically, your function's signature is

```
string matmul(const VI& d)
```

where `d` is an $(n+1)$-element vector such that the i^{th} matrix has dimensions $\texttt{d[i]} \times \texttt{d[i+1]}$.

The number of matrices, n satisfies $3 \leq n \leq 1000$, and each matrix dimension $\texttt{d[i]}$ satisfies $1 \leq \texttt{d[i]} \leq 1000$.