CSE100 Algorithm Design and Analysis Lab #8
Spring semester 2013 Miguel Á. Carreira-Perpiñán

# Dynamic Programming: Matrix Chain Multiplication

**Description** In this assignment you have to implement a dynamic programming algorithm: matrix chain multiplication (chapter 15.2), where the goal is to find the most computationally friendly matrix order when multiplying an arbitrary number of matrices in a row. You can assume that the entire input will be given as integers that can be stored using the standard C++ `int` type and that matrix sizes will be at least 1.

### Questions and input structure

1. **Number of operations**. Solve the matrix-chain multiplication dynamic programming algorithm and print the minimum number of operations required for each test case given. The first number, $n$, in the test case will tell you how many matrices are in the sequence ($n$ will not be negative). When the first number of a test case is 0 (zero), your program should terminate. The first number will be followed by $n+1$ numbers (unless $n$ was 0), indicating the size of the dimensions of the matrices. Your program should handle multiple test cases. After each test case, your program should output the lowest number of operations required to perform the matrix chain multiplication given in the test case. See the sample input/output.
   Testcase number to use with `grade_me`: 47.

2. **Printing the matrix sequence**. As for question 1, but print the matrix sequence with the correct parentheses. Each matrix should be named A#, where # is the matrix number starting at 0 (zero) and ending at $n-1$. See the sample input/output.
   Testcase number to use with `grade_me`: 48.

### Bonus questions

1. Compute the minimum number of operations but using the recursive version of the algorithm without memoization (section 15.3: Recursive-Matrix-Chain).

2. Compute the minimum number of operations but using the recursive version of the algorithm with memoization (section 15.3: Memoized-Matrix-Chain).

(Testcase number to use with `grade_me`: 47 in both cases.) Create a test input for a sequence of $n = 20$ matrices. How long do the (three) different versions of the algorithm take?

### Examples of input and output

*Input for Test Case 47*

```
2
2 3 5
3
10 100 5 50
3
10 30 5 60
6
```

```
30 35 15 5 10 20 25
0
```

*Output for Test Case 47*

```
30
7500
4500
15125
```

*Input for Test Case 48*

```
2
2 3 5
3
10 100 5 50
3
10 30 5 60
6
30 35 15 5 10 20 25
0
```

*Output for Test Case 48*

```
(A0A1)
((A0A1)A2)
((A0A1)A2)
((A0(A1A2))((A3A4)A5))
```

**Your solutions** Before leaving the lab, submit a zipped tar archive of your program through the assignments page of UCMCROPS. Please use your UCMNetID as the filename for the zipped tar archive. Be careful since UCMCROPS strictly enforces the assignment deadlines (deadlines will be every lab date at either 4:20pm or 7:20pm depending on your lab session.).