

AP COMPUTER SCIENCE A – PARSING FILE INPUT

The definition of matrix multiplication of two matrices **A** and **B**, for an $n \times m$ matrix **A** and an $m \times r$ matrix **B**, the matrix product $\mathbf{C} = \mathbf{AB}$ is given by the formula

$$c_{ik} = \sum_{j=1}^m a_{ij} b_{jk} .$$

Note that if the number of columns of **A** is not equal to the number of rows of **B**, then the matrix product is not defined. If the matrix product is defined, then matrix **C** is $n \times r$. In any case, a code segment which performs matrix multiplication is

```
int n, m, r;
double [][] a, b, c;
/**/
if ( a[0].length != b.length ) {
    System.out.println();
    System.out.println("The matrix product is not defined.");
    System.exit(0);
}
/**/
n=a.length;
m=a[0].length;
r=b[0].length;
c=new double [n][r];
/**/
for ( int i=0; i<n; ++i ) {
    for ( int k=0; k<r; ++k ) {
        c[i][k]=0.0;
        for ( int j=0; j<m; ++j ) c[i][k] += a[i][j]*b[j][k];
    }
}
```

Note that the code looks like the summation formula above.

The file `ab1.txt`, shown here, defines a 4×3 matrix **A** and a 3×2 matrix **B**. Here the matrix product $\mathbf{C} = \mathbf{AB}$ is defined with **C** being 4×2 . The format of this file is called *comma delimited*.

```
4 , 3
-9.0 , 10.0 , 2.0
3.0 , -16.0 , 4.0
2.0 , 8.0 , -10.0
12.0 , 2.0 , 17.0
3 , 2
-3.0 , -19.0
17.0 , -1.0
23.0 , 6.0
```

Similarly, the comma delimited file `ab2.txt`, shown here, defines a 4×3 matrix **A** and a 2×3 matrix **B**. Here the matrix product $\mathbf{C} = \mathbf{AB}$ is not defined.

```
4 , 3
-9.0 , 10.0 , 2.0
3.0 , -16.0 , 4.0
2.0 , 8.0 , -10.0
12.0 , 2.0 , 17.0
2 , 3
-3.0 , -19.0 , 58.0
17.0 , -1.0 , -10.0
```

AP COMPUTER SCIENCE A – PARSING FILE INPUT

If a program has the methods same methods as in `MaxSumTest`, *i.e.*, `open`, `readLine` and `close`, then each line of a comma delimited file can be read in as a `String []` by the method `readLineTokens` shown here.

```

/*
 * Read a comma delimited line as an array of tokens.
 */
private static String [] readLineTokens ( BufferedReader br ) {
    /**/
    String line;
    String [] tokens;
    /**/
    line=readLine(br);
    tokens=line.split(",");
    for ( int i=0; i<tokens.length; ++i ) tokens[i]=tokens[i].trim();
    /**/
    return tokens;
}

```

Alternatively, the matrices **A** and **B** can be defined with the *space delimited* files `ab3.txt`

```

4 3
-9.0  10.0  2.0
 3.0 -16.0  4.0
 2.0   8.0 -10.0
12.0   2.0 17.0
3 2
-3.0 -19.0
17.0 -1.0
23.0  6.0

```

and `ab4.txt`

```

4 3
-9.0  10.0  2.0
 3.0 -16.0  4.0
 2.0   8.0 -10.0
12.0   2.0 17.0
2 3
-3.0 -19.0 58.0
17.0 -1.0 -10.0

```

which files contain the same information as in, respectively, files `ab1.txt` and `ab2.txt` above.

AP COMPUTER SCIENCE A – PARSING FILE INPUT

If one wants to read in a line from a space delimited file as a `String []`, then `readLineTokens` from above needs to be modified as shown here.

```

/*
 * Read a space delimited line as an array of tokens.
 */
private static String [] readLineTokens ( BufferedReader br ) {
    /**/
    int numTokens, numNonEmptyTokens, k;
    String line;
    String [] tokens, rv;
    /**/
    line=readLine(br);
    tokens=line.split(" ");
    numTokens=tokens.length;
    /**/
    numNonEmptyTokens=0;
    for ( int i=0; i<numTokens; ++i ) {
        if ( !tokens[i].equals("") ) ++numNonEmptyTokens;
    }
    /**/
    rv=new String [numNonEmptyTokens];
    k=0;
    for ( int i=0; i<numTokens; ++i ) {
        if ( !tokens[i].equals("") ) rv[k++]=tokens[i];
    }
    /**/
    return rv;
}

```

Note that this method removes the empty tokens from the `String []`, similar to method `tokenizer2` in program `StringTokenizer`.

- 1) Starting with the file `MatrixMultiply1Stub.java`, make a program `MatrixMultiply1`, which can read in the files `ab1.txt` or `ab2.txt`, by:
 - a) Adding the appropriate code to method `multiply`.
 - b) Adding the appropriate code to method `readLineTokens`.
 - c) Adding the needed code to `readMatrix`. Follow the directions in the comments.
 - d) Adding the needed code to `printMatrix`. Follow the directions in the comments.

When running your program with `ab1.txt` as input, the output to the console should look like

```

243.0    173.0
-189.0   -17.0
-100.0  -106.0
389.0   -128.0

```

and with `ab2.txt` as input, like

The matrix product is not defined.

AP COMPUTER SCIENCE A – PARSING FILE INPUT

- 2) Make a program `MatrixMultiply2` by modifying `MatrixMultiply1` so that it can read in files `ab3.txt` or `ab4.txt`. You will have to change method `readLineTokens` appropriately.