

```

private boolean checkAvailable(int[] arr, int row, int col) {
    int len = arr.length;

    boolean res = (len == row * col ? true : false);

    if(res) {
        this.row = row;
        this.col = col;
    }

    return res;
}

private boolean checkAvailable(int[][] arr, int num) {
    int row = arr.length;
    int col = arr[0].length;
    int len = row * col;
    // System.out.printf("[ ] row = %d, col = %d\n", row, col);

    /*
    if(len % row == 0) {
        this.row = row;
        this.col = len / row;
    } else {
        System.out.printf("행렬로 변환할 수 없습니다.\n");
        System.out.printf("올바른 차원을 입력하세요.\n");
        System.out.printf("혹은 적절한 숫자(행)를 입력하세요\n");
        return false;
    }
    */
private boolean checkAvailable(int[] arr, int row) {
    int len = arr.length;

    /*
    if(len % row == 0) {
        this.row = row;
        this.col = len / row;
    } else {
        System.out.printf("행렬로 변환할 수 없습니다.\n");
        System.out.printf("올바른 차원을 입력하세요.\n");
        System.out.printf("혹은 적절한 숫자(행)를 입력하세요\n");
        return false;
    }
    */

    return DivideElement(len, row);
}

```

Q: 이 3가지 `boolean`이 왜 들어갔는지 모르겠다.  
 한가지 유추해볼만한 것은 첫번째 `boolean`은 Matrix A에 적용되고  
 두번째는 B와 C에 적용되는것 세번째는 `Wrong`에 적용된다는 사실까지는 이해했다.

```

private boolean DivideElement(int len, int row) {
    if(len % row == 0) {
        this.row = row;
        this.col = len / row;
    } else {
        System.out.printf("행렬로 변환할 수 없습니다.\n");
        System.out.printf("올바른 차원을 입력하세요.\n");
        System.out.printf("혹은 적절한 숫자(행)를 입력하세요\n");
        return false;
    }

    return true;
}

```

Q: 2, 3번째 `Checkavailable boolean`의 return 값으로 출력되는 `boolean`인데  
`len % row == 0` 을 하는 것은 `n * n` 배열의 행과 열이 알맞게 떨어지는지를 확인하기 위해서 ?

```

public boolean checkDimension(Matrix mat) {
    int row = mat.getRow();
    int col = mat.getCol();

    return (this.row == row) && (this.col == col);
}

public boolean checkDimension(Matrix A, Matrix B) {
    int Arow = A.getRow();
    int Brow = B.getRow();
    int Acol = A.getCol();
    int Bcol = B.getCol();

    return (Arow == Brow) && (Acol == Bcol);
}

```

Q: `CheckDimension` 전반적으로 무슨 동작을 하는 코드인지 왜 필요한지 잘 모르겠다.

```

ew int[row][col];

3      ==>   3 by 4
3 4      1 2 3
6 8      4 2 4
9 12     6 8 3
        6 9 12

allen = row * col;
mp = new int[totalLen];

    System.out.println("arr.length = " + arr.length);
    System.out.println("arr[0].length = " + arr[0].length);
    for(int i = 0; i < arr.length; i++) {
        (int j = 0; j < arr[0].length; j++) {
            tmp[i * arr[0].length + j] = arr[i][j];
            System.out.printf("tmp[%d] = %d\n",
                i * arr[0].length + j,
                tmp[i * arr[0].length + j]);
        }
    }

    for(int i = 0; i < row; i++) {
        (int j = 0; j < col; j++) {
            mat[i][j] = tmp[i * col + j];
        }
    }
}

```

Q: Matrix C에 해당하는 값을 출력하는 코드인데 {1, 2, 3, 4}. 이 왜 1 2 3  
 {3, 6, 9, 12} 6 8 3  
 6 9 12 로  
 3\*4에서 4\*3으로 변형돼서 출력되는지  
 정확히 이해하지 못했다.