```
public class Solution {
private int[][] grid = new int[][]{
                {8, 02, 22, 97, 38, 15, 00, 40, 00, 75, 04, 05, 07, 78, 52, 12, 50, 77, 91, 8},
                {49, 49, 99, 40, 17, 81, 18, 57, 60, 87, 17, 40, 98, 43, 69, 48, 04, 56, 62, 00},
                {81, 49, 31, 73, 55, 79, 14, 29, 93, 71, 40, 67, 53, 88, 30, 03, 49, 13, 36, 65},
                {52, 70, 95, 23, 04, 60, 11, 42, 69, 24, 68, 56, 01, 32, 56, 71, 37, 02, 36, 91},
                {22, 31, 16, 71, 51, 67, 63, 89, 41, 92, 36, 54, 22, 40, 40, 28, 66, 33, 13, 80},
                {24, 47, 32, 60, 99, 03, 45, 02, 44, 75, 33, 53, 78, 36, 84, 20, 35, 17, 12, 50},
                {32, 98, 81, 28, 64, 23, 67, 10, 26, 38, 40, 67, 59, 54, 70, 66, 18, 38, 64, 70},
                {67, 26, 20, 68, 02, 62, 12, 20, 95, 63, 94, 39, 63, 8, 40, 91, 66, 49, 94, 21},
                {24, 55, 58, 05, 66, 73, 99, 26, 97, 17, 78, 78, 96, 83, 14, 88, 34, 89, 63, 72},
                {21, 36, 23, 9, 75, 00, 76, 44, 20, 45, 35, 14, 00, 61, 33, 97, 34, 31, 33, 95},
                {78, 17, 53, 28, 22, 75, 31, 67, 15, 94, 03, 80, 04, 62, 16, 14, 90, 53, 56, 92},
                {16, 39, 05, 42, 96, 35, 31, 47, 55, 58, 88, 24, 00, 17, 54, 24, 36, 29, 85, 57},
                {86, 56, 00, 48, 35, 71, 89, 07, 05, 44, 44, 37, 44, 60, 21, 58, 51, 54, 17, 58},
                {19, 80, 81, 68, 05, 94, 47, 69, 28, 73, 92, 13, 86, 52, 17, 77, 04, 89, 55, 40},
                {04, 52, 8, 83, 97, 35, 99, 16, 07, 97, 57, 32, 16, 26, 26, 79, 33, 27, 98, 66},
                {88, 36, 68, 87, 57, 62, 20, 72, 03, 46, 33, 67, 46, 55, 12, 32, 63, 93, 53, 69},
                {04, 42, 16, 73, 38, 25, 39, 11, 24, 94, 72, 18, 8, 46, 29, 32, 40, 62, 76, 36},
                {20, 69, 36, 41, 72, 30, 23, 88, 34, 62, 99, 69, 82, 67, 59, 85, 74, 04, 36, 16},
                {20, 73, 35, 29, 78, 31, 90, 01, 74, 31, 49, 71, 48, 86, 81, 16, 23, 57, 05, 54},
                {01, 70, 54, 71, 83, 51, 54, 69, 16, 92, 33, 48, 61, 43, 52, 01, 89, 19, 67, 48}
        }; //made the grid from Project Euler #11, but made the program so any grid AxA can be
evaluated the same

        public Solution(){
                System.out.println(getProduct());
        }

        public int getProduct(){ //brings all the methods together and finds the greatest product
from them
                int product = diagonalProduct(this.grid);
                if(product <= verticalProduct(this.grid)){
                        product = verticalProduct(this.grid);
                }
                if(product <= horizontalProduct(this.grid)){
                        product = horizontalProduct(this.grid);
                }
                return product;
        }


        public int diagonalProduct(int[][] grid){ //checks both right and left diagonals
                int i = 0;
```

```
        int product = 0;
        int temp;
        while(i <= grid.length - 4){ //this while loop deals with columns
                for (int j = 0; j < grid.length - 4; ++j){ //right diagonal
                        temp = grid[i][j] * grid[i+1][j+1] * grid[i+2][j+2] * grid[i+3][j+3];
                        if(temp >= product){
                                product = temp;
                        }
                }
                for(int j = 3; j < grid.length; ++j){ //left diagonal
                        temp = grid[i][j] * grid[i+1][j-1] * grid[i+2][j-2] * grid[i+3][j-3];
                        if(temp >= product){
                                product = temp;
                        }
                }
                ++i;
        }
        return product;
}

public int horizontalProduct(int[][] grid){ //horizontal products of all combinations
        int product = 0;
        for(int i = 0; i < grid.length; ++i){
                for(int j = 0; j < grid.length-4; ++j){
                        if(grid[i][j] * grid[i][j+1] * grid[i][j+2] * grid[i][j+3] >= product){
                                product = grid[i][j] * grid[i][j+1] * grid[i][j+1] * grid[i][j+1];
                        }
                }
        }
        return product;
}
public int verticalProduct(int[][] grid){ //vertical products of all combinations
        int product = 0;
        for(int i = 0; i < grid.length-4; ++i){
                for(int j = 0; j < grid.length-1; ++j){
                        if(grid[i][j] * grid[i+1][j] * grid[i+2][j] * grid[i+3][j] >= product){
                                product = grid[i][j] * grid[i][j+1] * grid[i][j+1] * grid[i][j+1];
                        }
                }
        }
        return product;
}

public static void main(String[] args){
```

```
                Solution gridProd = new Solution();
        }
}
```