

## CS3430 Assignment 10

5 points

### What to Implement

A matrix is a 2D array of numbers. In linear algebra, binary matrix addition is an operation of taking two  $n \times n$  matrices **A** and **B** and computing a new  $n \times n$  matrix **C** such that  $C[r,c] = A[r,c] + B[r,c]$ , where  $1 \leq r \leq n$  and  $1 \leq c \leq n$ . Similarly, binary matrix subtraction is an operation of taking two  $n \times n$  matrices **A** and **B** and computing a new  $n \times n$  matrix **C** such that  $C[r,c] = A[r,c] - B[r,c]$ . For example, suppose that **A** and **B** are:

**A** =  $\begin{bmatrix} 10 & 11 \\ 13 & 14 \end{bmatrix}$

**B** =  $\begin{bmatrix} 1 & 2 \\ 4 & 5 \end{bmatrix}$

Then  $\mathbf{A} + \mathbf{B} = \mathbf{C}$ , where **C** is

$\begin{bmatrix} 11 & 13 \\ 17 & 19 \end{bmatrix}$

$\mathbf{A} - \mathbf{B} = \mathbf{D}$ , where **D** is

$\begin{bmatrix} 9 & 9 \\ 9 & 9 \end{bmatrix}$

Implement two Perl subroutines, `matrix_add` and `matrix_subtract`, to do binary matrix addition and subtraction and test your implementation with the following driver script.

```
#!/usr/bin/perl
```

```
use strict;  
use warnings;
```

```
sub matrix_add {  
    ## your Perl code goes here  
}
```

```
sub matrix_subtract {  
    ## your Perl code goes here.  
}
```

## print cells of a 2D matrix

```
sub print_matrix {  
    foreach (@_) {  
        foreach (@{$_}) {  
            print "$_ ";  
        }  
        print "\n";  
    }  
}
```

#####

# Tests

#####

```
my @m2by2_1 = (  
    [1, 2],  
    [4, 5]  
);
```

```
my @m2by2_2 = (  
    [10, 11],  
    [13, 14],  
);
```

```
my @m3by3_1 = (  
    [1, 2, 3],  
    [4, 5, 6],  
    [7, 8, 9]  
);
```

```
my @m3by3_2 = (  
    [10, 11, 12],  
    [13, 14, 15],
```

```

    [16, 17, 18]
);

my @m4by4_1 = (
    [1, 2, 3, 4],
    [5, 6, 7, 8],
    [9, 10, 11, 12],
    [13, 14, 15, 16],
);

my @m4by4_2 = (
    [10, 11, 12, 13],
    [14, 15, 16, 17],
    [18, 19, 20, 21],
    [22, 23, 24, 25],
);

print "Matrix Addition Results:\n\n";
my @add_m2by2 = matrix_add(\@m2by2_1, \@m2by2_2);
my @add_m3by3 = matrix_add(\@m3by3_1, \@m3by3_2);
my @add_m4by4 = matrix_add(\@m4by4_1, \@m4by4_2);

print_matrix(@add_m2by2); print "\n";
print_matrix(@add_m3by3); print "\n";
print_matrix(@add_m4by4); print "\n";

print "Matrix Subtraction Results:\n\n";
my @subt_m2by2 = matrix_subtract(\@m2by2_2, \@m2by2_1);
my @subt_m3by3 = matrix_subtract(\@m3by3_2, \@m3by3_1);
my @subt_m4by4 = matrix_subtract(\@m4by4_2, \@m4by4_1);

print_matrix(@subt_m2by2); print "\n";
print_matrix(@subt_m3by3); print "\n";
print_matrix(@subt_m4by4); print "\n";

```

After you implement `matrix_add` and `matrix_subtract` and run your script, it should produce the following output:

**Matrix Addition Results:**

**11 13**

**17 19**

**11 13 15**

**17 19 21**

**23 25 27**

**11 13 15 17**

**19 21 23 25**

**27 29 31 33**

**35 37 39 41**

**Matrix Subtraction Results:**

**9 9**

**9 9**

**9 9 9**

**9 9 9**

**9 9 9**

**9 9 9 9**

**9 9 9 9**

**9 9 9 9**

**9 9 9 9**

Create a .pl file and zip it up as HW10\_firstname\_lastname.zip