Exercise 8. Answer Sheet

Student's Name: _____Tran Thi Thoa_____   Student's ID: ___s1242006____

Problem 1. Write pseudo-code for the Strassen's algorithm.

```
def Strassen(A,B)
        n = A.rows
        C = new (nXn) matrix
        If n == 1
              Return A * B
        else
                compute A11, A12, A21, A22, B11, B12, B21, B22 which have the size m = n/2
                S1 = B12 - B22
                S2 = A11 + A12
                S3 = A21 + A22
                S4 = B21 - B11
                S5 = A11 + A22
                S6 = B11 + B22
                S7 = A12 - A22
                S8 = B21 + B22
                S9 = A11 - A21
                S10 = B11 + B12
                P1 = Strassen(A11, S1)
                P2 = Strassen(S2, B22)
                P3 = Strassen(S3, B11)
                P4 = Strassen(A22, S4)
                P5 = Strassen(S5, S6)
                P6 = Strassen(S7, S8)
                P7 = Strassen(S9,S10)
                C11 = P5 + P4 - P2 + P6
                C12 = P1 + P2
                C21 = P3 + P4
                C22 = P1 + P5 - P3 - P7
        end if
        return C
```

Problem 2. Use Strassen's algorithm to compute the matrix product:

$$\begin{pmatrix} 1 & 3 \\ 7 & 5 \end{pmatrix}\begin{pmatrix} 6 & 8 \\ 4 & 2 \end{pmatrix}$$

Show your work below:

We have A11 = [1]   B11 = [6]
A12 = [3]   B12 = [8]
A21 = [7]   B21 = [4]
A22 = [5]   B22 = [2]

Then S1 = B12 - B22 = [6]
S2 = A11 + A12 = [4]
S3 = A21 + A22 = [12]
S4 = B21 - B11 = [-2]
S5 = A11 + A22 = [6]
S6 = B11 + B22 = [8]
S7 = A12 - A22 = [-2]
S8 = B21 + B22 = [6]
S9 = A11 - A21 = [-6]
S10 = B11 + B12 = [14]

P1 = A11.S1 = [6]
P2 = S2.B22 = [8]
P3 = S3.B11 = [72]
P4 = A22.S4 = [-10]
P5 = S5.S6 = [48]
P6 = S7.S8 = [-12]
P7 = S9.S10 = [-84]

Therefore C11 = P5 + P4 - P2 + P6 = [18]
C12 = P1 + P2 = [14]
C21 = P3 + P4 = [62]
C22 = P1 + P5 - P3 - P7 = [66]

Return

$$C = \begin{bmatrix} 18 & 14 \\ 62 & 66 \end{bmatrix}$$

Problem 3. Make two programs implementing the Recursive matrix multiplication and the Strassen's algorithm. Upload your code. Generate two random matrices A and B of size n×n, multiply them using your programs and measure the time needed to get the result. Fill the following table:

Time needed to multiply two n×n matrices. (May depend on the programming language, computer, etc.)

Algorithm	n					
	32	64	128	256	512	1024
Recursive (sec)	22	165	1242	9816	77409	634310
Strassen (sec)	19	136	895	6266	43724	315238

Matrix multiplication using recursive algorithm and Strassen's algorithm implemented in C++
To compile and run the program, run the following command lines:

```
g++ -std=c++11 -o matrixmultiplication matrixmultiplication.cpp
```

./matrixmultiplication

Output is the time using those 2 algorithms for the matrix with the size 32, 42, 128, 256, 512 respectively

week8 — -bash — 80×24

```
                  ^
2 warnings and 1 error generated.
(base) wlan-napt-003:week8 thoatran$ g++ -std=c++11 -o matrixmultiplication matr
ixmultiplication.cpp
(base) wlan-napt-003:week8 thoatran$ ./matrixmultiplication
N = 32
Recursive Multiply takes: 22
Strassen Multiply takes: 19
N = 64
Recursive Multiply takes: 165
Strassen Multiply takes: 136
N = 128
Recursive Multiply takes: 1242
Strassen Multiply takes: 895
N = 256
Recursive Multiply takes: 9816
Strassen Multiply takes: 6266
N = 512
Recursive Multiply takes: 77409
Strassen Multiply takes: 43724
N = 1024
Recursive Multiply takes: 634310
Strassen Multiply takes: 315238
(base) wlan-napt-003:week8 thoatran$
```