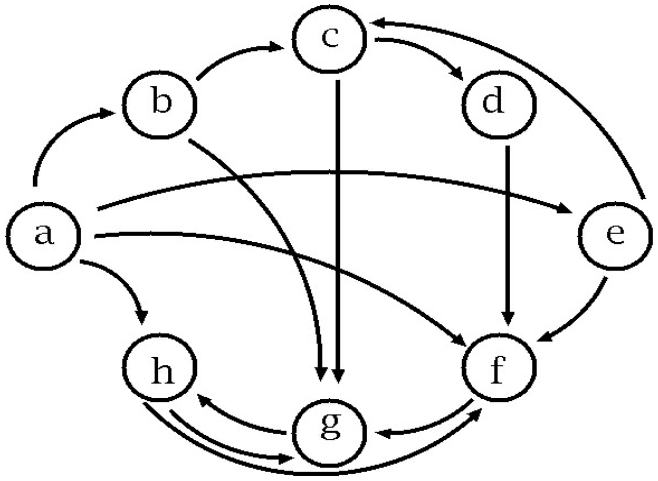
Exercise 6. Answer Sheet

Student's Name: Haruki Terushima Student's ID: s1290037

Problem 1. Given the graph below



a) *(10 points)* Fill the following matrix by putting 1 if there is an edge between nodes. Put 0 otherwise.

	a	b	c	d	e	f	g	h
a	1	1	0	0	1	1	0	1
b	0	1	1	0	0	0	1	0
c	0	0	1	1	0	0	1	0
d	0	0	0	1	0	1	0	0
e	0	0	1	0	1	1	0	0
f	0	0	0	0	0	1	1	0
g	0	0	0	0	0	0	1	1
h	0	0	0	0	0	1	1	1

b) *(40 points)* Write a program implementing Warshal's algorithm. Upload your code. Use your program to create a transitive closure G* of the graph above and show it in the space below.

Transitive closure defined by adjacency table

	a	b	c	d	e	f	g	h
a	1	1	1	1	1	1	1	1
b	0	1	1	1	0	1	1	1
c	0	0	1	1	0	1	1	1
d	0	0	0	1	0	1	1	1
e	0	0	1	1	1	1	1	1
f	0	0	0	0	0	1	1	1
g	0	0	0	0	0	1	1	1
h	0	0	0	0	0	1	1	1

<compile>
javac *.java
java Floyd < input

<input>

```
8
23
0 0
0 1
0 4
0 5
0 7
1 1
1 2
1 6
2 2
2 3
2 6
3 3
3 5
4 2
4 4
4 5
5 5
5 6
6 6
6 7
7 5
7 6
7 7
```

<output>
This is the first weight

1 1 0 0 1 1 0 1
0 1 1 0 0 0 1 0
0 0 1 1 0 0 1 0
0 0 0 1 0 1 0 0
0 0 1 0 1 1 0 0
0 0 0 0 0 1 1 0
0 0 0 0 0 0 1 1
0 0 0 0 0 1 1 1

This is the final distances using Floyd's Algorithm

1 1 1 1 1 1 1 1
0 1 1 1 0 1 1 1
0 0 1 1 0 1 1 1
0 0 0 1 0 1 1 1
0 0 1 1 1 1 1 1
0 0 0 0 0 1 1 1
0 0 0 0 0 1 1 1
0 0 0 0 0 1 1 1

Problem 2. *(50 points)* Consider the following weight adjacency matrix.

	a	b	c	d	e	f	g	h
a	0	48	∞	8	20	∞	20	∞
b	∞	0	24	∞	9	∞	76	29
c	97	∞	0	∞	∞	∞	18	1
d	∞	52	34	0	29	∞	∞	∞
e	∞	∞	∞	∞	0	10	∞	∞
f	∞	10	85	43	∞	0	41	29
g	∞	∞	∞	76	38	∞	0	∞
h	28	42	∞	77	21	∞	11	0

Write a program implementing Floyd's algorithm. Upload your code. Given the matrix above, calculate all pairs shortest paths using your program and fill the table below:

All pairs shortest path table

	a	b	c	d	e	f	g	h
a	0	40	42	8	20	30	20	43
b	53	0	24	61	9	19	42	25
c	29	22	0	37	22	12	18	1
d	63	49	34	0	29	39	52	35
e	67	20	44	53	0	10	51	39
f	57	10	34	43	19	0	41	29
g	105	58	82	76	38	48	0	77
h	28	21	45	36	21	11	48	0

<compile>
javac *.java
java Floyd < input

<input>

```
8
27
0 1 48
0 3 8
0 4 20
0 6 20
1 2 24
1 4 9
1 6 76
1 7 29
2 0 97
```

```
2 6 18
2 7 1
3 1 52
3 2 34
3 4 29
4 5 10
5 1 10
5 2 85
5 3 43
5 6 41
5 7 29
6 3 76
6 4 38
7 0 28
7 1 42
7 3 77
7 4 21
7 5 11
```

<output>
This is the first weight

0 48 inf 8 20 inf 20 inf
inf 0 24 inf 9 inf 76 29
97 inf 0 inf inf inf 18 1
inf 52 34 0 29 inf inf inf
inf inf inf inf 0 10 inf inf
inf 10 85 43 inf 0 41 29
inf inf inf 76 38 inf 0 inf
28 42 inf 77 21 11 inf 0

This is the final distances using Floyd's Algorithm

0 40 42 8 20 30 20 43
53 0 24 61 9 19 42 25
29 22 0 37 22 12 18 1
63 49 34 0 29 39 52 35
67 20 44 53 0 10 51 39
57 10 34 43 19 0 41 29
105 58 82 76 38 48 0 77
28 21 45 36 21 11 48 0