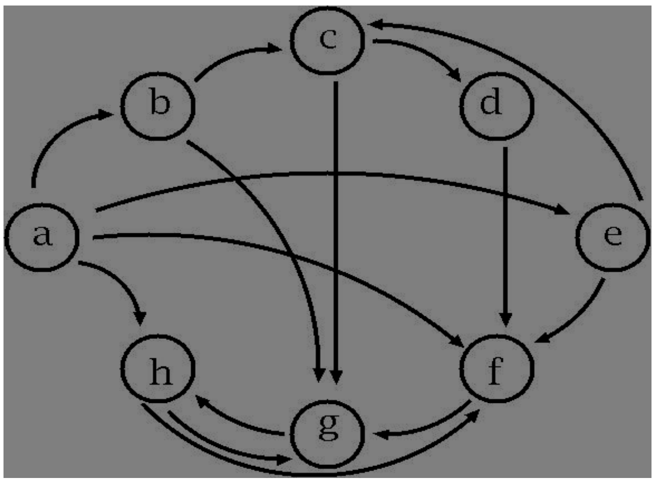
Exercise 6. Answer Sheet

Student's Name: Tomonori Masubuchi Student's ID: s1240078

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

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	0	1	1	1	1	1	1	1
b	0	0	1	1	0	1	1	1
c	0	0	0	1	0	1	1	1
d	0	0	0	0	0	1	1	1
e	0	0	1	1	0	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

```
#include<stdio.h>

int n;
int func[100][100];


int main(){
  int i,j,k;

  printf("Please input:");
  scanf("%d",&n);

  for(i = 0;i < n;i ++)
    {
      for(j = 0;j < n;j ++)
      {
        scanf("%d",&func[i][j]);
      }
    }


  for(k = 0;k < n;k ++)
    {
      for(i = 0;i < n;i ++)
      {
        for(j = 0;j < n;j ++)
          {
            if(func[i][j] != 1)func[i][j] = func[i][k] * func[k][j];
          }
      }
    }

  printf("\nOutput\n");

  for(i = 0;i < n;i ++)
    {
      for(j = 0;j < n;j ++)
```

```
          {
            printf("%d ",func[i][j]);
          }
          printf("\n");
        }

      return 0;

    }
```

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	36	25
c	29	42	0	37	22	32	12	1
d	63	49	34	0	29	39	46	35
e	67	20	44	53	0	10	50	39
f	57	10	34	43	19	0	40	29
g	105	58	82	76	38	48	0	77
h	28	41	65	36	21	31	11	0

```
#include<stdio.h>
#include<stdlib.h>
#include<string.h>

#define INF 99999

int min(int x, int y){
  if(x == 0) return x;
  if(x == INF || y < x) return y;
  else return x;
}

int n;
int func[100][100];

int main(){

  int i,j,k;

  char c[100];

  printf("Please input: ");
  scanf("%d", &n);

  for(i = 0;i < n;i ++)
    {
      for(j = 0;j < n;j ++)
            {
              scanf("%s", c);

              if(strcmp(c,"infinity") == 0)func[i][j] = INF;
              else func[i][j] = atoi(c);
            }
    }

  for(k = 0;k < n;k ++)
    {
      for(i = 0;i < n;i ++)
            {
              for(j = 0;j < n;j ++)
                {
                  if(func[i][k] != INF && func[k][j] != INF) func[i][j] = min(func[i][j], func[i][k] + func[k][j]);
                }
            }
    }

  printf("\nOutput\n");

  for(i = 0;i < n;i ++)
    {
      for(j = 0;j < n;j ++)
            {
              if(func[i][j] == INF)printf("infinity ");
              else printf("%d ",func[i][j]);
            }
      printf("\n");
    }
  return 0;
}
```