```c
//01
//2020118008 박보경
//본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.

#define _CRT_SECURE_NO_WARNINGS

#include<stdio.h>
#include<stdlib.h>
#define COMPARE(x,y) (((x)<(y))? -1 : ((x))==((y))? 0 : 1 )

#define MAX_TERMS 100

typedef struct
{
	float coef;
	int exp;
}term;
term terms[MAX_TERMS];
int avail = 0;

void inputPoly(char* fname, int* startA, int* finishA, int* startB, int* finishB);//파일로부터 데이터를 입력받아 2개의 다항식을 구성
void printPoly(int start, int finish); //하나의 다항식을 전달받아 화면으로 출력
void padd(int startA, int finishA, int startB, int finishB, int* startD, int* finishD); //2.6 두 함수의 poly를 더하는 함수
void attach(float coefficient, int exponent); //다항식에 새 항 추가하는 함수

int main()
{
	char* filename = "input.txt";
	int startA, startB, finishA, finishB, startD, finishD;
	inputPoly(filename, &startA, &finishA, &startB, &finishB);
	printf("two input polynomials ... \n");
	printf("A(x) = ");

	printf("  \n");

	printPoly(startA, finishA);
	printf("B(x) = ");
	printPoly(startB, finishB);

	printf("  \n \n");
```

```
	printf("the result of polynomial addition ...\n");
	printf("D(x) = A(x) + B(x) = \n");
	padd(startA, finishA, startB, finishB, &startD, &finishD);
	printPoly(startD, finishD);
	return 0;
}


void inputPoly(char* fname, int* startA, int* finishA, int* startB, int* finishB) //파일로부터 데이터를 입력받아 2개의 다항식을 구성
{
	int Asize, Bsize, expp;
	float coeff;
	FILE* fp = fopen(fname, "r");
	fscanf_s(fp, "%d %d", &Asize, &Bsize);
	*startA = 0;
	*finishA = Asize - 1;
	*startB = Asize;
	*finishB = (Asize + Bsize)-1;
	avail = (Asize + Bsize);

	for (int i = 0; i < avail; i++)
	{
		fscanf_s(fp, "%f %d ", &coeff, &expp);
		terms[i].coef = coeff;
		terms[i].exp = expp;
	}
}

void printPoly(int start, int finish)
{
	for (int i = start; i <= finish; i++)
	{
		if (i == finish)
		{
			if (terms[i].exp == 0)
			{
				printf("%.1f", terms[i].coef);
			}
			else
			{
				printf("%.1fx^%d ", terms[i].coef, terms[i].exp);
			}
```

```
                }
                else
                {
                        printf("%.1fx^%d ", terms[i].coef, terms[i].exp);
                        printf("+ ");
                }
        }
}
void padd(int startA, int finishA, int startB, int finishB, int* startD, int* finishD) //2.6 두 함수의 poly를 더하는 함수
{ /* A(x) + B(x) = D(x) */

        float coefficient;
        *startD = avail;

        while (startA <= finishA && startB <= finishB)
        {
                switch (COMPARE(terms[startA].exp, terms[startB].exp))
                {
                case -1: /* a expon < b expon */
                        attach(terms[startB].coef, terms[startB].exp);
                        startB++;
                        break;
                case 0: /* equal exponents */
                        coefficient = terms[startA].coef + terms[startB].coef;
                        if (coefficient)
                                attach(coefficient, terms[startA].exp);
                        startA++;
                        startB++;
                        break;
                case 1: /* a expon > b expon */
                        attach(terms[startA].coef, terms[startA].exp);
                        startA++;
                }
        }
        /* add in remaining terms of A(x) */
        for (; startA <= finishA; startA++)
                attach(terms[startA].coef, terms[startA].exp);

        /* add in remaining terms of B(x) */
        for (; startB <= finishB; finishB++)
                attach(terms[startB].coef, terms[startB].exp);
```

```
        *finishD = avail - 1;
}

void attach(float coefficient, int exponent) //다항식에 새 항 추가하는 함수
{
        if (avail >= MAX_TERMS)
        {
                fprintf(stderr, "Too may terms in the polynomial\n");
                exit(EXIT_FAILURE);
        }
        terms[avail].coef = coefficient;
        terms[avail++].exp = exponent;
}



//02
//2020118008 박보경
//본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.

#define _CRT_SECURE_NO_WARNINGS

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

#define MAX_TERMS 100

typedef struct
{
        int row;
        int col;
        int value;
}term;

int avail = 9;

void inputmatrix(term *A,char* fname);
void transpose(term a[], term b[]);
void printmatrix(term c[]);

int main()
{
        term A[MAX_TERMS];
        term B[MAX_TERMS];
```

```
char* filename1 = "a.txt";
char* filename2 = "b.txt";

int ary[7][6] = { 0 };
int bry[7][6] = { 0 };


inputmatrix(A,filename1);


int i, j;
for (i = 1; i <= A->value; i++)
{
	ary[A[i].row][A[i].col] = A[i].value;
}


transpose(A,B);


for (i = 1; i <= B->value; i++)
{
	bry[B[i].row][B[i].col] = B[i].value;
}

printf("A \n");
for (i = 0; i < 7; i++)
{
	for (j = 0; j < 6; j++)
	{
		printf("%d ", ary[i][j]);
	}
	printf("\n");
}

printf("\n");

printf("B \n");
for (i = 0; i < 7; i++)
{
	for (j = 0; j < 6; j++)
	{
		printf("%d ", bry[i][j]);
```

```
                }
                printf("\n");
        }


        return 0;
}


void inputmatrix(term *A,char* fname) //파일로부터 희소행렬을 입력받음
{

        FILE* fp = fopen(fname, "r");

        int row2, col2, value2;

        for (int i = 0; i < avail; i++)
        {
                fscanf_s(fp, "%d %d %d ", &row2, &col2, &value2);
                A[i].row = row2;
                A[i].col = col2;
                A[i].value = value2;
        }
}

void transpose(term a[], term b[])
{
        int n, i, j, currentb;
        n = a[0].value;
        b[0].row = a[0].col;
        b[0].col = a[0].row;
        b[0].value = n;

        if (n > 0)
        {
                currentb = 1;
                for(i=0 ; i < a[0].col ; i++)
                        for(j=1; j<=n; j++)
                                if (a[j].col == i)
                                {
                                        b[currentb].row = a[j].col;
                                        b[currentb].col = a[j].row;
                                        b[currentb].value = a[j].value;
```

```
                                        currentb++;
                                }
        }
}

void printmatrix(term c[])
{
        for (int i = 0; i < avail; i++)
        {
                printf("%3d %3d %3d  \n",c[i].row, c[i].col, c[i].value);
        }
}

//02 함수로 printmatrix한 소스코드
//2020118008 박보경
//본인은 이 소스파일을 다른 사람의 소스를 복사하지 않고 직접 작성하였습니다.

#define _CRT_SECURE_NO_WARNINGS

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

#define MAX_TERMS 100

typedef struct
{
        int row;
        int col;
        int value;
}term;

int avail = 9;

void inputmatrix(term* A, char* fname);
void transpose(term a[], term b[]);
void printmatrix(term* C, int (*ary)[6]);
void inputFile(term* D, char* fname);

int main()
{
        term A[MAX_TERMS];
        term B[MAX_TERMS];
        char* filename1 = "a.txt";
```

```
	char* filename2 = "b.txt";

	int ary[7][6] = { 0 };
	int bry[7][6] = { 0 };


	inputmatrix(A, filename1);

	transpose(A, B);

	printf("A \n");
	printmatrix(A, ary);

	printf("\n");

	printf("B \n");
	printmatrix(B, bry);


	char* filenameB = "b.txt";
	inputFile(B, filenameB);


	return 0;
}


void inputmatrix(term* A, char* fname) //파일로부터 희소행렬을 입력받음
{

	FILE* fp = fopen(fname, "r");

	int row2, col2, value2;

	for (int i = 0; i < avail; i++)
	{
		fscanf_s(fp, "%d %d %d ", &row2, &col2, &value2);
		A[i].row = row2;
		A[i].col = col2;
		A[i].value = value2;
	}
}
```

```
void transpose(term a[], term b[])
{
	int n, i, j, currentb;
	n = a[0].value;
	b[0].row = a[0].col;
	b[0].col = a[0].row;
	b[0].value = n;

	if (n > 0)
	{
		currentb = 1;
		for (i = 0; i < a[0].col; i++)
			for (j = 1; j <= n; j++)
				if (a[j].col == i)
				{
					b[currentb].row = a[j].col;
					b[currentb].col = a[j].row;
					b[currentb].value = a[j].value;
					currentb++;
				}
	}
}

void printmatrix(term *C, int (*ary)[6])
{
	for (int i = 1; i <= C->value; i++)
	{
		ary[C[i].row][C[i].col] = C[i].value;
	}

	for (int i = 0; i < 7; i++)
	{
		for (int j = 0; j < 6; j++)
		{
			printf("%d ", ary[i][j]);
		}
		printf("\n");
	}
}


void inputFile(term* D, char* fname)
{
```

```
        FILE* fp = fopen(fname, "w");

        for (int i = 0; i < avail; i++)
        {
                fprintf(fp, "%d %d %d\n", D[i].row, D[i].col, D[i].value);

        }

}
```