```c
#include <stdio.h>

int makeT(int n, int A[n][n]);
void removeRepeated(int n, int A[n][n]);
int differentT(int n, int A[n][n], int first, int second);
void addUP(int n, int A[n][n], int first, int second);

int main()
{
    int A[11][11] = {
        { 0, 2, 3, 1, 0, 0, 0, 0, 0, 0, 0 },
        { 2, 0, 0, 0, 2, 0, 1, 0, 0, 0, 0 },
        { 2, 0, 0, 0, 7, 4, 0, 0, 0, 0, 0 },
        { 1, 0, 0, 0, 0, 3, 5, 0, 0, 0, 0 },
        { 0, 2, 7, 0, 0, 0, 0, 4, 5, 0, 0 },
        { 0, 0, 4, 3, 0, 0, 0, 6, 0, 2, 0 },
        { 0, 1, 0, 5, 0, 0, 0, 0, 3, 3, 0 },
        { 0, 0, 0, 0, 4, 6, 0, 0, 0, 0, 7 },
        { 0, 0, 0, 0, 5, 0, 3, 0, 0, 0, 4 },
        { 0, 0, 0, 0, 0, 2, 3, 0, 0, 0, 4 },
        { 0, 0, 0, 0, 0, 0, 0, 7, 4, 4, 0 }
    };

    removeRepeated(11, A);

    printf("\nVerticles sorted by weight:");


    for (int i = 1; i <= 7; i++)
    {
        printf("\n%d: ", i);

        for (int j = 1; j <= 11; j++)
        {
            for (int k = 1; k <= 11; k++)
            {
                if (A[j - 1][k - 1] == i)
                {
                    printf("%d-%d; ", j, k);
                }
            }
        }
    }


    int B[11][11];
    makeT(11, B);

    printf("\n\nOur path: ");

    for (int i = 1; i <= 7; i++)
    {
        for (int j = 1; j <= 11; j++)
        {
            for (int k = 1; k <= 11; k++)
            {
                if (A[j - 1][k - 1] == i && differentT(11, B, j, k))
                {
                    addUp(11, B, j, k);
                    printf("%d-%d; ", j, k);
                }
            }
        }
    }
    printf("\n\n");

    return 0;
}

int makeT(int n, int A[n][n])
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            A[i][j] = 0;
        }
    }
    for (int i = 0; i < n; i++)
    {
        A[i][i] = i + 1;
    }

    return A[n][n];
}

void removeRepeated(int n, int A[n][n])
{
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (j < i)
            {
                A[i][j] = 0;
            }
        }
    }
}

int differentT(int n, int A[n][n], int first, int second)
{
    int temp1;
    int temp2;

    // line
    for (int i = 0; i < n; i++)
    {
        temp1 = 0;
        temp2 = 0;
        // first element
        for (int j = 0; j < n; j++)
        {
            if (A[i][j] == first)
            {
                temp1 = 1;
            }
        }
        // second element
        for (int k = 0; k < n; k++)
        {
            if (A[i][k] == second)
            {
                temp2 = 1;
            }
        }

        if (temp1 && temp2)
        {
            return 0;
        }
    }

    return 1;
}

void addUp(int n, int A[n][n], int first, int second)
{
    int scndLine;
    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (A[i][j] == second)
            {
                scndLine = i;
            }
        }
    }

    for (int i = 0; i < n; i++)
    {
        for (int j = 0; j < n; j++)
        {
            if (A[i][j] == first)
            {
                for (int k = 0; k < n; k++)
                {
                    if (A[scndLine][k])
                    {
                        A[i][k] = A[scndLine][k];
                        A[scndLine][k] = 0;
                    }
                }
            }
        }
    }
}
```