```
#define _CRT_SECURE_NO_WARNINGS

#include <stdio.h>

#define MAX 5
int main(void) {

	char source[] = { 'M','T','B','U','V' };
	char destination[] = { 'T','B','V','M','U' };
	int bus[] = { 244,178,54,141,50 };
	int minutes[] = { 120,90,60,75,55 };


	int i = 0,found=0, k=0;
	char c = ' ',insource, indest;
	int route[MAX] = { -1,-1,-1,-1,-1 };
	int currentroute=-1;
	printf("Source        Destination       Bus No.       minutes\n");
	while (i < MAX) {
		switch (source[i]) {
		case 'M':
			printf("Markham");
			break;
		case 'U':
			printf("Mississauga");
			break;
		case 'B':
			printf("Brampton");
			break;
		case 'V':
			printf("Vaungh");
			break;
		case 'T':
			printf("Toronto");
			break;
		}
	switch (destination[i]) {
		case 'M':
			printf("%8c Markham",c);
			break;
		case 'U':
			printf("%6c Mississauga",c);
			break;
		case 'B':
			printf("%8c Brampton",c);
			break;
		case 'V':
			printf("%8c Vaungh",c);
			break;
		case 'T':
			printf("%8c Toronto",c);
			break;
		}
	printf("%12d %10d", bus[i], minutes[i]);
	printf("\n");
	i = i + 1;


	}

	printf("Enter Source(M|V|U|B|T): ");
	scanf(" %c", &insource);
	printf("\nEnter Destination(M|V|U|B|T): ");
	scanf(" %c", &indest);
```

```
        printf("\n");
        printf("Direct Routes\n");
        for (i = 0; i < MAX; i++) {
                if (source[i] == insource && destination[i] == indest) {

                        printf("%d %4d\n", bus[i], minutes[i]);
                        found = 1;
                }

        }
        if (found == 0) {
                printf("No Direct route");
        }
        printf("\n\n");
        printf("Connected Routes\n");

        int temp = 0;
        found = 0;
        for (i = 0; i < MAX; i++) {
                if (source[i] == insource && destination[i] != indest) {
                        route[temp] = i;
                        temp++;
                        k = 0;
                        do {
                                currentroute=route[temp-1];
                                if (source[k]==destination[currentroute]) {
                                        route[temp] = k;
                                        temp++;
                                        if (destination[k] == indest) {
                                                found = 1;

                                        }
                                }
                                k=1;


                        } while (k < MAX || found != 1);
                }
        }
        if (found == 0) {
                printf("No Connected route");
        }
        else {
                for (i = 0; i < MAX && route[i]!=-1; i++) {
                        currentroute=route[i];
                        printf("%c %4d\n", bus[currentroute], minutes[currentroute]);
                }
        }
}
```