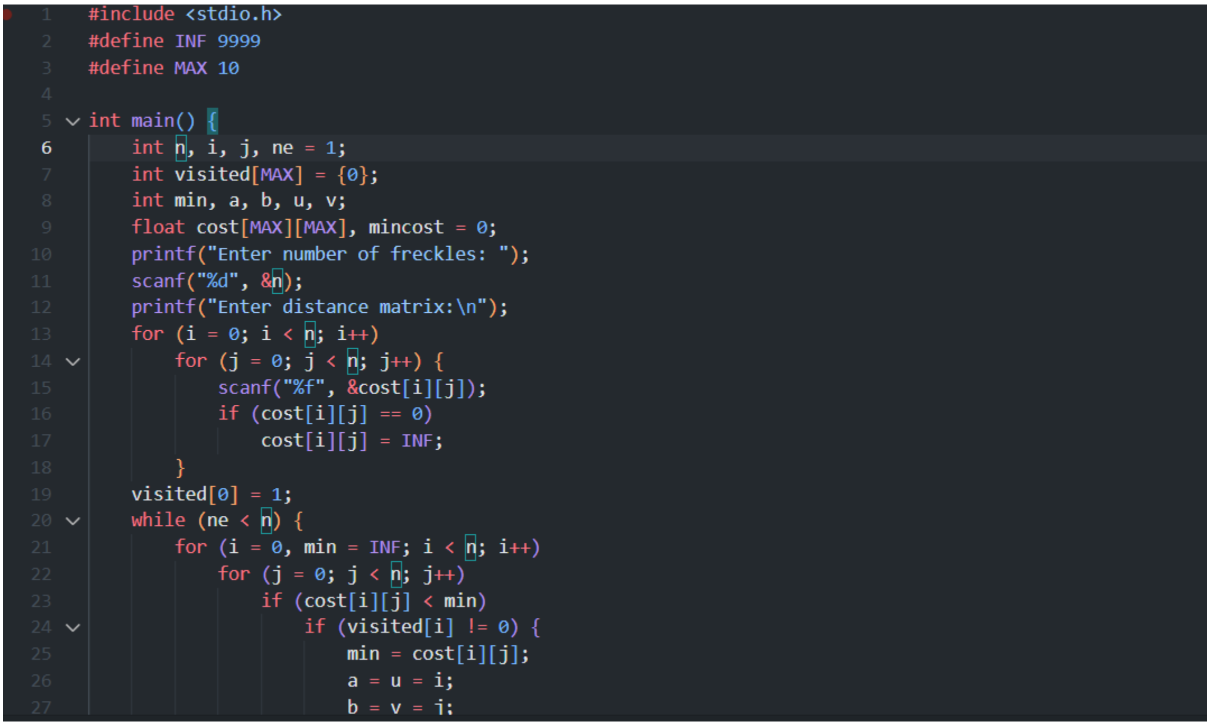
Name – Tanishq Mankari

Section – A8_B2

Roll number – 31

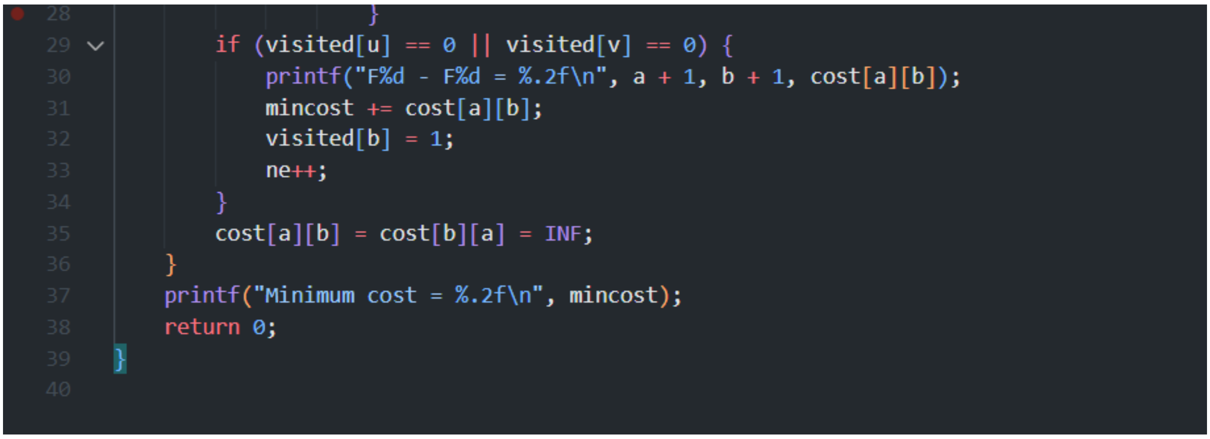
Practical – 2

Code – Part A



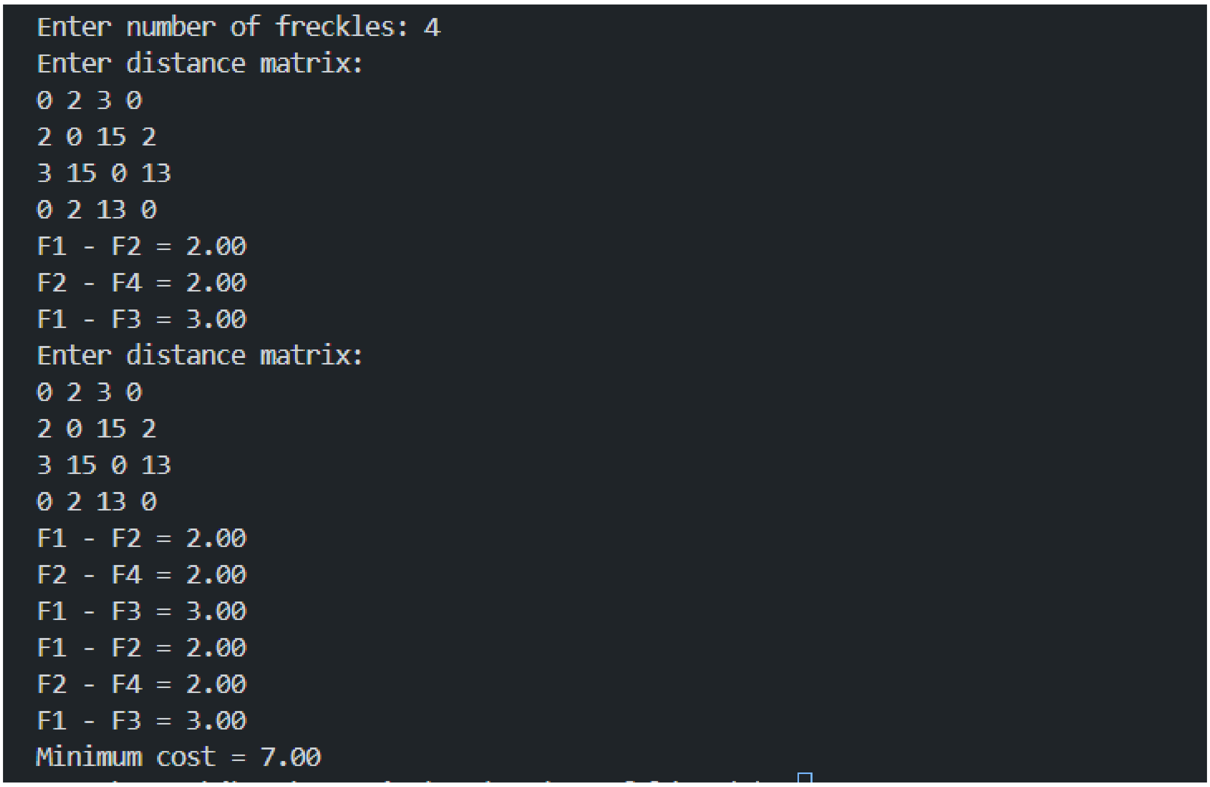
```c
#include <stdio.h>
#define INF 9999
#define MAX 10

int main() {
    int n, i, j, ne = 1;
    int visited[MAX] = {0};
    int min, a, b, u, v;
    float cost[MAX][MAX], mincost = 0;
    printf("Enter number of freckles: ");
    scanf("%d", &n);
    printf("Enter distance matrix:\n");
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++) {
            scanf("%f", &cost[i][j]);
            if (cost[i][j] == 0)
                cost[i][j] = INF;
        }
    visited[0] = 1;
    while (ne < n) {
        for (i = 0, min = INF; i < n; i++)
            for (j = 0; j < n; j++)
                if (cost[i][j] < min)
                    if (visited[i] != 0) {
                        min = cost[i][j];
                        a = u = i;
                        b = v = j;
```



```c
                    }
        if (visited[u] == 0 || visited[v] == 0) {
            printf("F%d - F%d = %.2f\n", a + 1, b + 1, cost[a][b]);
            mincost += cost[a][b];
            visited[b] = 1;
            ne++;
        }
        cost[a][b] = cost[b][a] = INF;
    }
    printf("Minimum cost = %.2f\n", mincost);
    return 0;
}
```

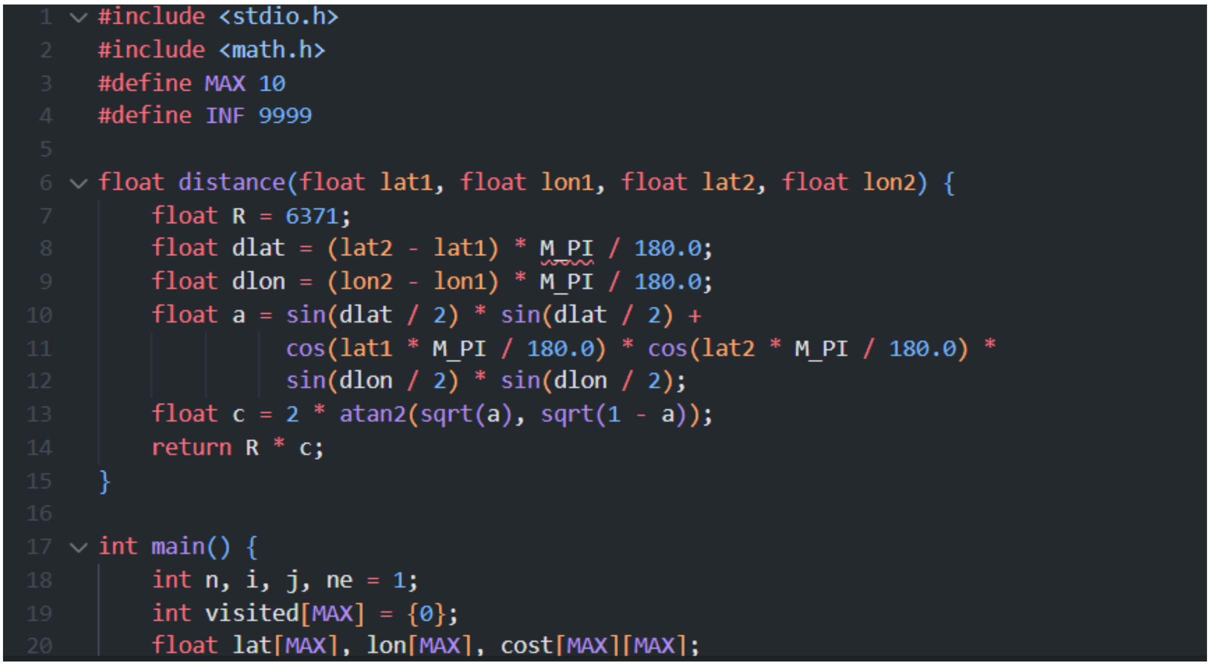
Output –



```
Enter number of freckles: 4
Enter distance matrix:
0 2 3 0
2 0 15 2
3 15 0 13
0 2 13 0
F1 - F2 = 2.00
F2 - F4 = 2.00
F1 - F3 = 3.00
Enter distance matrix:
0 2 3 0
2 0 15 2
3 15 0 13
0 2 13 0
F1 - F2 = 2.00
F2 - F4 = 2.00
F1 - F3 = 3.00
F1 - F2 = 2.00
F2 - F4 = 2.00
F1 - F3 = 3.00
Minimum cost = 7.00
```

Part B

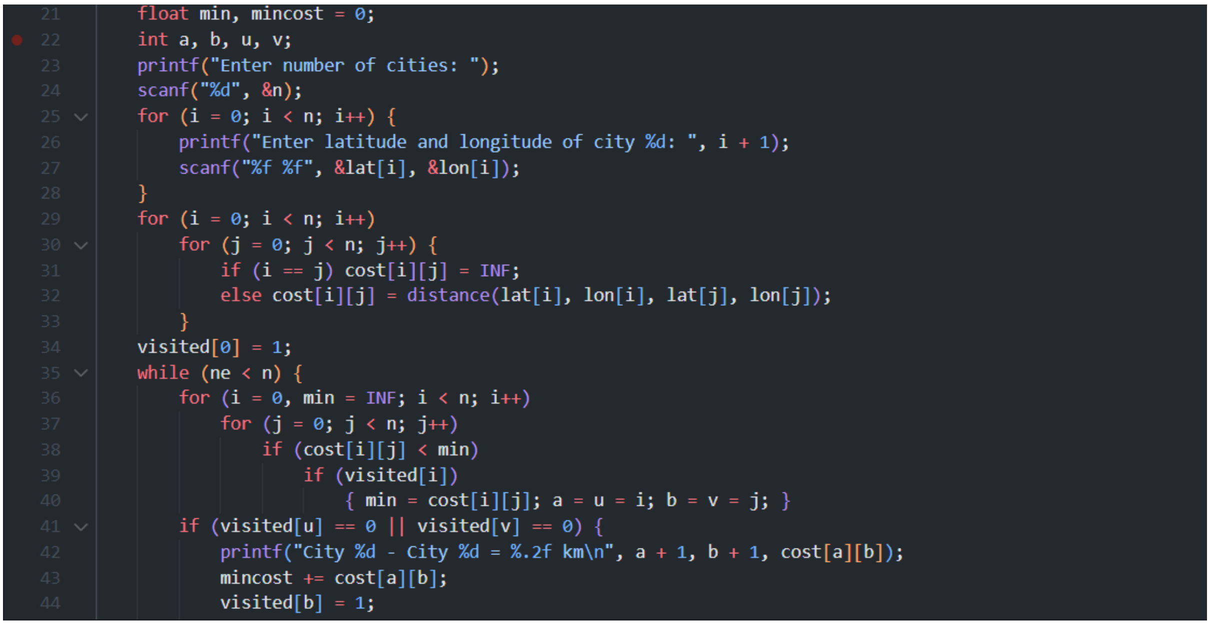
Code –



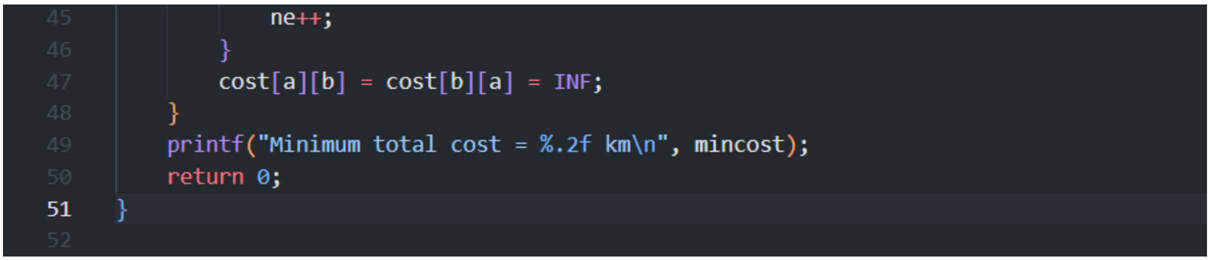
```c
#include <stdio.h>
#include <math.h>
#define MAX 10
#define INF 9999

float distance(float lat1, float lon1, float lat2, float lon2) {
    float R = 6371;
    float dlat = (lat2 - lat1) * M_PI / 180.0;
    float dlon = (lon2 - lon1) * M_PI / 180.0;
    float a = sin(dlat / 2) * sin(dlat / 2) +
              cos(lat1 * M_PI / 180.0) * cos(lat2 * M_PI / 180.0) *
              sin(dlon / 2) * sin(dlon / 2);
    float c = 2 * atan2(sqrt(a), sqrt(1 - a));
    return R * c;
}

int main() {
    int n, i, j, ne = 1;
    int visited[MAX] = {0};
    float lat[MAX], lon[MAX], cost[MAX][MAX];
```

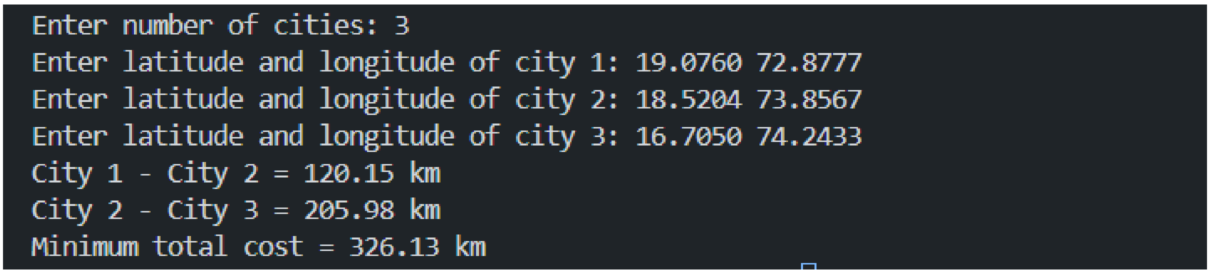


```
    float min, mincost = 0;
    int a, b, u, v;
    printf("Enter number of cities: ");
    scanf("%d", &n);
    for (i = 0; i < n; i++) {
        printf("Enter latitude and longitude of city %d: ", i + 1);
        scanf("%f %f", &lat[i], &lon[i]);
    }
    for (i = 0; i < n; i++)
        for (j = 0; j < n; j++) {
            if (i == j) cost[i][j] = INF;
            else cost[i][j] = distance(lat[i], lon[i], lat[j], lon[j]);
        }
    visited[0] = 1;
    while (ne < n) {
        for (i = 0, min = INF; i < n; i++)
            for (j = 0; j < n; j++)
                if (cost[i][j] < min)
                    if (visited[i])
                        { min = cost[i][j]; a = u = i; b = v = j; }
        if (visited[u] == 0 || visited[v] == 0) {
            printf("City %d - City %d = %.2f km\n", a + 1, b + 1, cost[a][b]);
            mincost += cost[a][b];
            visited[b] = 1;
```



```
            ne++;
        }
        cost[a][b] = cost[b][a] = INF;
    }
    printf("Minimum total cost = %.2f km\n", mincost);
    return 0;
}
```

Output –



```
Enter number of cities: 3
Enter latitude and longitude of city 1: 19.0760 72.8777
Enter latitude and longitude of city 2: 18.5204 73.8567
Enter latitude and longitude of city 3: 16.7050 74.2433
City 1 - City 2 = 120.15 km
City 2 - City 3 = 205.98 km
Minimum total cost = 326.13 km
```