```cpp
//Kalli Bonin and Brian Chang
//Question 3 - Water Taxi

#include <iostream>
#include <cmath>
#include <cstdlib>
#include <fstream>
#include <iomanip>

using namespace std;

int main()
{
    //declare price of trip
    const int SERVICE = 11;
    const double PER_KM = 2.7;

    ifstream fin("taxi.txt");
    ofstream fout("earnings.txt");

    //check if file opens
    if (!fin || !fout)
    {
        cout << "Could not open file.";
        return EXIT_FAILURE;
    }

    double totalDistance = 0, totalCost = 0;
    double longestTrip = 0, leastExpensive = 1e6;
    unsigned int numberTrip = 0;

    //set header for the table
    fout << fixed << setprecision(2)
         << setw(4)  << "Trip"
         << setw(8)  << "Return"
         << setw(7)  << "Stops"
         << setw(10) << "Distance"
         << setw(10) << "Cost"
         << setw(12) << "Cumulative"
         << setw(12) << "Cumulative" << endl;

    fout << setw(51) << "Distance"
         << setw(12) << "Cost" << endl;

    //while there is still data to read from the file
    while (fin.good())
    {
        //add one to the number of trips
        numberTrip++;
        //read in to see if customer makes a round trip
        bool roundTrip = 0;
        fin >> roundTrip;

        //read in number of stops
        unsigned int numberStops = 0;
        fin >> numberStops;

        //this makes sure it doesn't add an extra row of data that doesn't exist
        if (numberStops > 0)
        {
            double distance = 0;

            double lastX = 0, lastY = 0;

            //read in coordinates of stops and calculate distance
            for (int i = 0; i < numberStops; i++)
            {
                double curX = 0, curY = 0;
                fin >> curX >> curY;

                distance += sqrt(pow(lastX-curX, 2) + pow(lastY-curY,2));

                lastX = curX;
                lastY = curY;
            }

            //if the customer makes a round trip, add the last distance
            if (roundTrip)
                distance += sqrt(lastX*lastX + lastY*lastY);

            //calculate the cost per trip
            double cost = 0;
            cost = SERVICE*numberStops + distance*PER_KM;

            //add to cumulative
            totalDistance += distance;
            totalCost += cost;

            //check for record length and cost
            if (distance > longestTrip)
                longestTrip = distance;
            if (cost < leastExpensive)
                leastExpensive = cost;

            //output all numbers
            fout << setw(4)  << numberTrip
                 << setw(8)  << roundTrip
                 << setw(7)  << numberStops
                 << setw(10) << distance
                 << setw(10) << cost
                 << setw(12) << totalDistance
                 << setw(12) << totalCost << endl;
        }

    }

    //output totals and records
    fout << "Cumulative Distance: " << totalDistance << "km" << endl
         << "Cumulative Cost: $" << totalCost << endl
         << "Longest Trip: " << longestTrip << "km" << endl
```

```
          << "Least Expensive Trip: $" << leastExpensive << endl;

    //close file
    fin.close();
    fout.close();
}


```