

03b

O que faz o código a seguir (que continua na página ao lado)?

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

struct POI {
    char name[100];
    double lat;
    double lon;
};

double calcDistance(double lat1, double lon1, double lat2, double lon2) {
    const double R = 6371.0; // Earth's radius in km

    double dLat = (lat2 - lat1) * M_PI / 180.0;
    double dLon = (lon2 - lon1) * M_PI / 180.0;

    double a = sin(dLat / 2.0) * sin(dLat / 2.0) +
        cos(lat1 * M_PI / 180.0) * cos(lat2 * M_PI / 180.0) *
        sin(dLon / 2.0) * sin(dLon / 2.0);

    double c = 2.0 * atan2(sqrt(a), sqrt(1.0 - a));

    double distance = R * c;
    return distance;
}

struct POI* filterPoints(const struct POI poiArray[], int numPoints,
    const struct POI poi, double maxDistance,
    int* numSelected) {

    struct POI* selectedPOIs =
        (struct POI*) malloc(numPoints * sizeof(struct POI));

    if (!selectedPOIs) {
        *numSelected = 0;
        return NULL; // Memory allocation failed
    }

    int count = 0;
    for (int i = 0; i < numPoints; i++) {
        double d = calcDistance(poi.lat, poi.lon,
            poiArray[i].lat, poiArray[i].lon);
        if (d < maxDistance) {
            selectedPOIs[count] = poiArray[i];
            count++;
        }
    }

    return selectedPOIs;
}
```

```
    }
}
*numSelected = count;
return selectedPOIs;
}

int main() {
    struct POI poiArray[] = {
        {"Centro de Tecnologia", -29.713318, -53.71663},
        {"Biblioteca Central", -29.71566, -53.71523},
        {"Centro de Convenções", -29.72237, -53.71718},
        {"Planetário", -29.72027, -53.71726},
        {"Reitoria da UFSM", -29.72083, -53.71479},
        {"Restaurante Universitário 2", -29.71400, -53.71937},
        {"Hospital Universitário de Santa Maria", -29.71368, -53.71536},
        {"Pulsar Incubadora Tecnológica - Prédio 2", -29.71101, -53.71634},
        {"Pulsar Incubadora Tecnológica - Prédio 61H", -29.72468, -53.71335},
        {"Casa do Estudante Universitário - CEU II", -29.71801, -53.71465}};

    int numPoints = sizeof(poiArray) / sizeof(poiArray[0]);
    struct POI jbsm = {"Jardim Botânico", -29.71689, -53.72968};
    double maxDistance = 1.5;

    int numSelected;
    struct POI* selectedPOIs =
        filterPoints(poiArray, numPoints, jbsm, maxDistance, &numSelected);

    for (int i = 0; i < numSelected; i++) {
        printf("%s (%.6f, %.6f)\n",
            selectedPOIs[i].name, selectedPOIs[i].lat, selectedPOIs[i].lon);
    }

    free(selectedPOIs);

    return 0;
}
```

Este código é bem diferente do código 03a, mas há um padrão algorítmico de processamento que se repete nos 2 códigos. Vocês conseguem identificar qual é?