

Actividad de repaso - tercera parte

Programación 2

hector.pereira@estudiantes.utec.edu.uy

5.582.582-5

14/03/2025

Resumen

En esta actividad se realizan los siguientes objetivos:

- Se define un arreglo de 100 puntos el cual es cargado con puntos aleatorios.
- Suponiendo que los 100 puntos en el orden que están cargados en el arreglo forman un camino, se calcula la distancia total de dicho camino.
- Se guardan los cien puntos cargados en el arreglo en un archivo llamado "puntos.txt", escribiendo en cada línea las coordenadas x e y de cada punto.

Librerías

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

stdio.h: Librería para el uso de funciones básicas de entrada y salida.

stdlib.h: Implementada por sus funciones de redacción y escritura de archivos, además de la función "rand()" para la inicialización aleatoria del arreglo de puntos.

math.h: Utilizada por su función de raíz cuadrada "sqrt()" dentro del cálculo de distancias.

time.h: Implementada para asignar nuevas semillas a la función rand para evitar repeticiones de valores aleatorios.

Programa

```
struct Point {  
    int x;  
    int y;  
};
```

Estructura de dato de punto, donde se almacenan ambas coordenadas en el plano.

```
float distance(struct Point p1, struct Point p2) {  
    return sqrt((p1.x - p2.x) * (p1.x - p2.x) + (p1.y - p2.y) * (p1.y -  
    p2.y));  
}
```

Función de cálculo de distancia entre dos puntos utilizando el teorema de pitágoras.

```
int saveData(struct Point points[100]) {  
    FILE *file = fopen("puntos.txt", "w");  
    if (file == NULL) {  
        printf("Error abriendo archivo\n");  
        return 1;  
    }  
  
    for (int i = 0; i < 100; i++) {  
        fprintf(file, "%d %d\n", points[i].x, points[i].y);  
    }  
  
    printf("Distancia total: %.2f\n", totalDistance);  
    printf("Datos guardados en puntos.txt\n");  
    fclose(file);  
    return 0;  
}
```

Función dedicada para guardar datos dentro del archivo “puntos.txt”, con verificación de errores de apertura y confirmación de guardado.

```

int main() {
    srand(time(NULL));

    struct Point points[100];
    for (int i = 0; i < 100; i++) {
        points[i].x = rand() % 100;
        points[i].y = rand() % 100;
    }

    float totalDistance = 0;
    for (int i = 0; i < 99; i++) {
        totalDistance += distance(points[i], points[i + 1]);
    }
    printf("Distancia total: %.1f\n", totalDistance);

    saveData(points);

    return 0;
}

```

Función principal donde se asigna la semilla srand como el tiempo actual para mitigar repeticiones en la generación de numeros aleatorios de coordenadas. Luego se calcula la distancia total, para luego guardar los datos de las coordenadas de los puntos usando la función “saveData”.

Salidas

- Salida 1

```

Distancia total: 5245.51
Datos guardados en puntos.txt

```

- Salida 2

```

Distancia total: 5195.47
Datos guardados en puntos.txt

```

- Salida 3

```

Distancia total: 4956.03
Datos guardados en puntos.txt

```