Program implementasi pohon biner.

1. Pada awal program, kita mendefinisikan struktur Node yang memiliki tiga anggota: data (nilai node), left (pointer ke anak kiri), dan right (pointer ke anak kanan).
2. Fungsi **createNode()** digunakan untuk membuat node baru dengan nilai yang diberikan. Jika node baru tidak dapat dibuat, fungsi ini mengembalikan **NULL**.
3. Fungsi **insertNode()** digunakan untuk menambahkan node ke dalam pohon. Jika pohon masih kosong, maka node baru akan menjadi akar. Jika tidak, node akan ditempatkan sesuai dengan aturan pohon biner: nilai yang lebih kecil akan ditempatkan di anak kiri, dan nilai yang lebih besar akan ditempatkan di anak kanan.
4. Fungsi **searchNode()** digunakan untuk mencari elemen tertentu dalam pohon. Fungsi ini menggunakan rekursi untuk mencari nilai dalam pohon secara berulang hingga nilai ditemukan atau mencapai akhir pohon.
5. Fungsi **inorderTraversal()** digunakan untuk menampilkan elemen dalam pohon secara inorder (urutan kiri-akar-kanan). Fungsi ini juga menggunakan rekursi untuk mengunjungi setiap node dalam urutan yang benar.
6. Fungsi **displayMenu()** digunakan untuk menampilkan menu kepada pengguna. Menu ini memberikan opsi untuk menambahkan elemen ke dalam pohon, mencari elemen dalam pohon, menampilkan pohon secara inorder, dan keluar dari program.
7. Di dalam fungsi **main()**, kita membuat sebuah pointer **root** yang awalnya menunjuk ke **NULL** sebagai pohon awal.
8. Program menggunakan loop **do-while** untuk terus menampilkan menu kepada pengguna dan memproses pilihan menu yang diberikan.
9. Di dalam loop **do-while**, program menggunakan switch-case untuk memproses setiap pilihan menu yang dipilih oleh pengguna. Untuk setiap pilihan, program melakukan operasi yang sesuai, seperti menambahkan elemen ke dalam pohon, mencari elemen dalam pohon, menampilkan pohon secara inorder, atau keluar dari program.
10. Program akan terus berjalan hingga pengguna memilih opsi keluar (pilihan 4).