

Module 7: Pointers and Memory

The image displays two screenshots of an Online C++ 17 Compiler IDE, showing the implementation and execution of a binary search algorithm.

Top Screenshot: The code defines a `struct Item` with `string name;` and `int id;`. A `binarySearch` function is implemented, which takes an array of `Item` pointers, its size, and a search ID. It uses a while loop to find the item. The `main` function dynamically allocates an array of 100 `Item` objects, populates them with sorted IDs and names, and calls `binarySearch` to find the item with ID 95.

```
1 #include <iostream>
2 #include <string>
3 using namespace std;
4
5 struct Item {
6     string name;
7     int id;
8 };
9
10 // Binary Search Function
11 int binarySearch(Item* arr, int size, int searchId) {
12     int left = 0;
13     int right = size - 1;
14
15     while (left <= right) {
16         int mid = (left + right) / 2;
17
18         if (arr[mid].id == searchId) {
19             return mid;
20         } else if (arr[mid].id < searchId) {
21             left = mid + 1;
22         } else {
23             right = mid - 1;
24         }
25     }
26
27     return -1;
28 }
29
30 int main() {
31     int SIZE = 100;
32
33     // Dynamically allocate array
34     Item* inventory = new Item[SIZE];
35
36     // Populate sorted sample data
37     for (int i = 0; i < SIZE; i++) {
38         inventory[i].id = i + 1;
39     }
40 }
```

The output shows the search results: "Enter an ID to search (1-100): 95" and "Found! ID: 95, Name: Item_95". The execution time is 15.45 seconds.

Bottom Screenshot: The code is the same as the top screenshot, but the `main` function includes a prompt for the user to enter an ID to search. The output shows the search results: "Enter an ID to search (1-100): 95" and "Found! ID: 95, Name: Item_95". The execution time is 15.45 seconds.

```
24 }
25 }
26
27 return -1;
28 }
29
30 int main() {
31     int SIZE = 100;
32
33     // Dynamically allocate array
34     Item* inventory = new Item[SIZE];
35
36     // Populate sorted sample data
37     for (int i = 0; i < SIZE; i++) {
38         inventory[i].id = i + 1;
39         inventory[i].name = "Item_" + to_string(i + 1);
40     }
41
42     int searchId;
43     cout << "Enter an ID to search (1-100): ";
44     cin >> searchId;
45
46     // Perform binary search
47     int index = binarySearch(inventory, SIZE, searchId);
48
49     if (index != -1) {
50         cout << "Found! ID: " << inventory[index].id
51              << ", Name: " << inventory[index].name << endl;
52     } else {
53         cout << "Item with ID " << searchId << " not found." << endl;
54     }
55
56     // Free the memory
57     delete[] inventory;
58
59     return 0;
60 }
```

The output shows the search results: "Enter an ID to search (1-100): 95" and "Found! ID: 95, Name: Item_95". The execution time is 15.45 seconds.