add.c | day33.c | reverse.c | table.c | dobydo | break.c | fact.c | nam.c | oh.c | odd.c | oood.c

day33.c > ...

```c
#include <stdio.h>

int main() {
    int n;
    printf("Enter the number of elements: ");
    scanf("%d", &n);

    int arr[n];
    printf("Enter %d sorted elements:\n", n);
    for (int i = 0; i < n; i++) {
        scanf("%d", &arr[i]);
    }

    int key;
    printf("Enter the element to search: ");
    scanf("%d", &key);

    int low = 0, high = n - 1, mid;
    int index = -1; // default: not found

    while (low <= high) {
        mid = (low + high) / 2;

        if (arr[mid] == key) {
            index = mid;
            break;
        }
        else if (arr[mid] < key) {
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ACER\Desktop\itscbe> gcc day33.c
PS C:\Users\ACER\Desktop\itscbe> ./a.exe
Enter the number of elements: 4
Enter 4 sorted elements:
3 5
5
5
Enter the element to search: 5
Found at index 1
PS C:\Users\ACER\Desktop\itscbe>
```

0 ⚠ 0 Ln 44, Col 1 Spaces: 2 UTF-8 CRLF {} C Go Li

C 33.c × C reverse.c C table.c C dobydo C break.c C fact.c C nam.c C oh.c C odd.c C oood.c C lp.c

C 33.c > ...

```c
int main() {
    for (int i = 0; i < n; i++) {

    int key;
    printf("Enter the element to insert: ");
    scanf("%d", &key);

    int i;
    // Find position where key should be inserted
    for (i = n - 1; (i >= 0 && arr[i] > key); i--) {
        arr[i + 1] = arr[i]; // shift elements to right
    }
    arr[i + 1] = key; // insert key

    printf("Array after insertion:\n");
    for (i = 0; i <= n; i++) {
        printf("%d ", arr[i]);
    }
    printf("\n");

    return 0;
}
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
PS C:\Users\ACER\Desktop\itscbe> ./a.exe
Enter the number of elements: 3
Enter 3 sorted elements:
2
3
4
Enter the element to insert: 3
Array after insertion:
2 3 3 4
PS C:\Users\ACER\Desktop\itscbe>
```

pow pow pow pow pow pow pow pow

⊗ 0 ⚠ 0 Ln 33, Col 1 Spaces: 2 UTF-8 CRLF {} C Go Live