

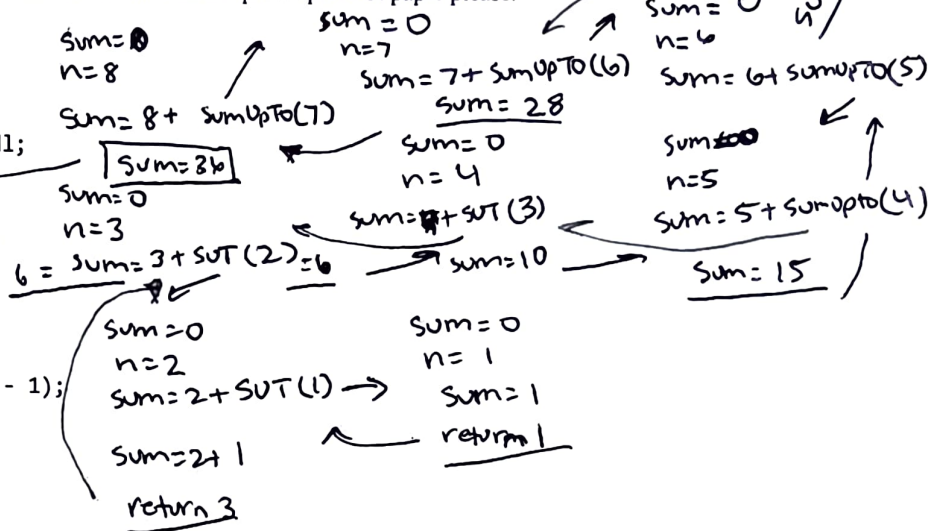
Provide a clear trace of the following recursive functions. Use a separate piece of paper please.

1. Sum

cout << sumUpTo(8) << endl;

Function *****
 OUTPUT: 36

```
int sumUpTo(int n) {
    int sum = 0;
    if (n == 1) sum = 1;
    else sum = n + sumUpTo(n - 1);
    return sum;
}
```



2. Binary Search

```
const int arraySize = 9;
int a[] = {1,6,7,12,15,22,26,41,50};
cout << binarySearch(a, 0, arraySize, 6) << endl;
```

Function *****

```
int binarySearch(const int anArray[], int first, int last, int target) {
    int index;
    if (first > last)
        index = -1;
    else {
        int mid = first + (last - first) / 2;
        if (target == anArray[mid])
            index = mid;
        else if (target < anArray[mid])
            index = binarySearch(anArray, first, mid - 1, target);
        else
            index = binarySearch(anArray, mid + 1, last, target);
    }
    return(index);
}
```

Handwritten trace of the binarySearch function:

- mid = 4
- 6 == 15 ⇒ no/false
- 6 < 15 ⇒ yes/true
- index = binarySearch(a, 0, 3, 6)
- mid = 1
- 6 == 6 ⇒ true
- index = 1
- return 1
- Final return value: 2

OUTPUT:
2