



Section 1 - Coding

Q1 Test Case Input Output

6
3 2 1 4 5 6

13

Weightage - 10 Input Output

7
10 2 7 4 5 8 1

25

Weightage - 20 Input Output

7
10 2 7 4 5 7 1

24

Weightage - 20 Input Output

20
19 20 17 18 16 15 12 13 11 14 10 1 8 2 3 9 4 5 6 7

114

Weightage - 25 Input Output

20
19 20 17 18 16 15 12 13 11 14 10 1 8 2 3 9 76 128 98 53
283

Weightage - 25 Sample Input Sample Output

5
2 7 9 3 1
12

Sample Input Sample Output

6
5 1 2 6 3 4
15

Solution

```
#include <iostream>
using namespace std;

const int MAX_N = 20; // Define a maximum size for the array

int maximumLoot(int hval[], int n) {
    if (n < 0) {
        return 0;
    }
    if (n == 0) {
        return hval[0];
    }

    int pick = hval[n] + maximumLoot(hval, n - 2);
    int notPick = maximumLoot(hval, n - 1);

    return max(pick, notPick);
}

int main() {
    int n;
    cin >> n;

    int hval[MAX_N];
    for (int i = 0; i < n; i++) {
        cin >> hval[i];
    }

    cout << maximumLoot(hval, n - 1);

    return 0;
}
```

```

#include <stdio.h>

#define MAX_N 20

int maximumLoot(int hval[], int n) {
    if (n < 0) {
        return 0;
    }
    if (n == 0) {
        return hval[0];
    }

    int pick = hval[n] + maximumLoot(hval, n - 2);
    int notPick = maximumLoot(hval, n - 1);

    return (pick > notPick) ? pick : notPick;
}

int main() {
    int n;
    scanf("%d", &n);

    int hval[MAX_N];
    for (int i = 0; i < n; i++) {
        scanf("%d", &hval[i]);
    }

    printf("%d", maximumLoot(hval, n - 1));

    return 0;
}

```

Q2 Test Case Input Output

5

Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total number of moves: 31

Weightage - 10 Input Output

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 6 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C

Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 7 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 5 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 6 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B

Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 8 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 6 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 5 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 7 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 6 from A to C

Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of moves: 255

Weightage - 40 Input Output

Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 6 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 4 from C to A
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 5 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B

Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 7 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 5 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 4 from C to A
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 6 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C

Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total number of moves: 127

Weightage - 20 Input Output

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 6 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C

Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of moves: 63

Weightage - 20 Input Output

4

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of moves: 15

Weightage - 10 Sample Input Sample Output

3

Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total number of moves: 7

Sample Input Sample Output

2

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of moves: 3

Solution

```

#include <stdio.h>

int countMoves(int n, char source, char auxiliary, char destination) {
    if (n == 1) {
        printf("Move disk 1 from %c to %c\n", source, destination);
        return 1;
    }

    int moves = 0;
    moves += countMoves(n - 1, source, destination, auxiliary); // Move n-1 disks
from source to auxiliary
    printf("Move disk %d from %c to %c\n", n, source, destination);
    moves += 1; // Move the largest disk from source to destination
    moves += countMoves(n - 1, auxiliary, source, destination); // Move n-1 disks
from auxiliary to destination

    return moves;
}

int main() {
    int n;
    scanf("%d", &n);

    char source = 'A';
    char auxiliary = 'B';
    char destination = 'C';

    int totalMoves = countMoves(n, source, auxiliary, destination);
    printf("Total number of moves: %d\n", totalMoves);

    return 0;
}

```

```

#include <iostream>
using namespace std;

int countMoves(int n, char source, char auxiliary, char destination) {
    if (n == 1) {
        cout << "Move disk 1 from " << source << " to " << destination << endl;
        return 1;
    }

    int moves = 0;
    moves += countMoves(n - 1, source, destination, auxiliary); // Move n-1 disks
    from source to auxiliary
    cout << "Move disk " << n << " from " << source << " to " << destination <<
endl;
    moves += 1; // Move the largest disk from source to destination
    moves += countMoves(n - 1, auxiliary, source, destination); // Move n-1 disks
    from auxiliary to destination

    return moves;
}

int main() {
    int n;
    cin >> n;

    char source = 'A';
    char auxiliary = 'B';
    char destination = 'C';

    int totalMoves = countMoves(n, source, auxiliary, destination);
    cout << "Total number of moves: " << totalMoves<<endl;

    return 0;
}

```

Q3 Test Case Input Output

5

31

Weightage - 10 Input Output

8

255

Weightage - 40 Input Output

7

127

Weightage - 20 Input Output

6

63

Weightage - 20 Input Output

4

15

Weightage - 10 Sample Input Sample Output

2

3

Sample Input Sample Output

4

15

Solution

```
#include <stdio.h>

int towerOfHanoiMoves(int n) {
    if (n == 0) {
        return 0;
    }
    return 2 * towerOfHanoiMoves(n - 1) + 1;
}

int main() {
    int n;
    scanf("%d", &n);

    int moves = towerOfHanoiMoves(n);
    printf("%d", moves);

    return 0;
}
```

```

#include <iostream>
using namespace std;

int towerOfHanoiMoves(int n) {
    if (n == 0) {
        return 0;
    }
    return 2 * towerOfHanoiMoves(n - 1) + 1;
}

int main() {
    int n;
    cin >> n;

    int moves = towerOfHanoiMoves(n);
    cout << moves ;

    return 0;
}

```

Q4 Test Case Input Output

1
20
-1

Weightage - 10 Input Output

3
1 2 2
2

Weightage - 10 Input Output

10
1 2 3 4 5 6 7 8 9 9
9

Weightage - 15 Input Output

15
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15
-1

Weightage - 15 Input Output

20
1 2 3 4 5 6 7 8 9 10 11 11 12 12 13 13 14 14 15 15
11 12 13 14 15

Weightage - 25 Input Output

25

1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

-1

Weightage - 25 Sample Input Sample Output

4

0 1 3 2

-1

Sample Input Sample Output

5

1 1 2 2 3

1 2

Solution

```

#include <iostream>
using namespace std;

// Function to find and print duplicate elements in a sorted array
int findAndPrintDuplicates(int arr[], int n, int prev) {
    // Base case: If the array has only one element, it cannot have duplicates
    if (n == 1) {
        if (arr[0] == prev) {
            cout << arr[0] << " ";
            return arr[0];
        }
        return -1;
    }

    // Recursive case
    int current = arr[n - 1];
    int result = findAndPrintDuplicates(arr, n - 1, current);

    // Check if the current element is a duplicate
    if (current == prev) {
        cout << current << " ";
        return current;
    }

    return result;
}

int main() {
    int n;
    //cout << "Enter the number of elements: ";
    cin >> n;

    int arr[n];

    //cout << "Enter " << n << " sorted elements: ";
    for (int i = 0; i < n; i++) {
        cin >> arr[i];
    }

    int result = findAndPrintDuplicates(arr, n, -1);

    if (result == -1) {
        cout << "-1";
    }

    return 0;
}

```

```

#include <stdio.h>

// Function to find and print duplicate elements in a sorted array
int findAndPrintDuplicates(int arr[], int n, int prev) {
    // Base case: If the array has only one element, it cannot have duplicates
    if (n == 1) {
        if (arr[0] == prev) {
            printf("%d ", arr[0]);
            return arr[0];
        }
        return -1;
    }

    // Recursive case
    int current = arr[n - 1];
    int result = findAndPrintDuplicates(arr, n - 1, current);

    // Check if the current element is a duplicate
    if (current == prev) {
        printf("%d ", current);
        return current;
    }

    return result;
}

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

    int arr[n];

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

    int result = findAndPrintDuplicates(arr, n, -1);

    if (result == -1) {
        printf("-1");
    }

    return 0;
}

```

Q5 Test Case Input Output

```

2
3 5
6 5

243
7776

```

Weightage - 10 Input Output

2
1 5
5 5

1
3125

Weightage - 10 Input Output

1
102 5

11040808032

Weightage - 15 Input Output

1
200 5

3200000000000

Weightage - 15 Input Output

3
65 5
15 5
25 2

1160290625
759375
625

Weightage - 25 Input Output

3
5 5
7 4
185 5

3125
2401
216699865625

Weightage - 25 Sample Input Sample Output

2
15 5
100 5

759375
100000000000

Sample Input Sample Output

1
7 5

16807

Solution

```
#include <stdio.h>

long long power(long long a, long long b) {
    if (b == 0) {
        return 1;
    } else if (b % 2 == 0) {
        long long half_pow = power(a, b / 2);
        return (half_pow * half_pow);
    } else {
        long long half_pow = power(a, (b - 1) / 2);
        return (a * half_pow * half_pow);
    }
}

int main() {
    int T;
    scanf("%d", &T);

    while (T--) {
        long long a, b;
        scanf("%lld %lld", &a, &b);

        long long result = power(a, b);
        printf("%lld\n", result);
    }

    return 0;
}
```

```

#include <iostream>
using namespace std;

long long power(long long a, long long b) {
    if (b == 0) {
        return 1;
    } else if (b % 2 == 0) {
        long long half_pow = power(a, b / 2);
        return (half_pow * half_pow);
    } else {
        long long half_pow = power(a, (b - 1) / 2);
        return (a * half_pow * half_pow);
    }
}

int main() {
    int T;
    cin >> T;

    while (T--) {
        long long a, b;
        cin >> a >> b;

        long long result = power(a, b);
        cout << result << endl;
    }

    return 0;
}

```

Q6 Test Case Input Output

5


```
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total number of times the disk is moved: 31
```

Weightage - 10 Input Output

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 6 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C

Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 7 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 5 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 6 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B

Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 8 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 6 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 5 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 7 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 6 from A to C

Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of times the disk is moved: 255

Weightage - 40 Input Output

Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 6 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 4 from C to A
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 5 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B

Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 7 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 5 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 4 from C to A
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 6 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 4 from A to B
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 3 from C to B
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 5 from A to C

Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Move disk 3 from B to A
Move disk 1 from C to B
Move disk 2 from C to A
Move disk 1 from B to A
Move disk 4 from B to C
Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total number of times the disk is moved: 127

Weightage - 20 Input Output

6

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 5 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 4 from C to B
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 6 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 4 from B to A
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 3 from C to A
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 5 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C

Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of times the disk is moved: 63

Weightage - 20 Input Output

4

Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of times the disk is moved: 15

Weightage - 10 Sample Input Sample Output

3

Move disk 1 from A to C
Move disk 2 from A to B
Move disk 1 from C to B
Move disk 3 from A to C
Move disk 1 from B to A
Move disk 2 from B to C
Move disk 1 from A to C
Total number of times the disk is moved: 7

Sample Input Sample Output

4

```
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Move disk 3 from A to B
Move disk 1 from C to A
Move disk 2 from C to B
Move disk 1 from A to B
Move disk 4 from A to C
Move disk 1 from B to C
Move disk 2 from B to A
Move disk 1 from C to A
Move disk 3 from B to C
Move disk 1 from A to B
Move disk 2 from A to C
Move disk 1 from B to C
Total number of times the disk is moved: 15
```

Solution

```
#include <stdio.h>

void towerOfHanoiMoves(int n, char source, char destination, char auxiliary, int
*total_moves) {
    if (n == 0) {
        return;
    }

    // Move (n-1) disks from source to auxiliary
    towerOfHanoiMoves(n - 1, source, auxiliary, destination, total_moves);

    // Move the nth disk from source to destination
    printf("Move disk %d from %c to %c\n", n, source, destination);
    (*total_moves)++;

    // Move the (n-1) disks from auxiliary to destination
    towerOfHanoiMoves(n - 1, auxiliary, destination, source, total_moves);
}

int main() {
    int n;
    scanf("%d", &n);

    int total_moves = 0;
    towerOfHanoiMoves(n, 'A', 'C', 'B', &total_moves);

    printf("Total number of times the disk is moved: %d\n", total_moves);

    return 0;
}
```

```

#include <iostream>
using namespace std;

void towerOfHanoiMoves(int n, char source, char destination, char auxiliary, int
&total_moves) {
    if (n == 0) {
        return;
    }

    // Move (n-1) disks from source to auxiliary
    towerOfHanoiMoves(n - 1, source, auxiliary, destination, total_moves);

    // Move the nth disk from source to destination
    cout << "Move disk " << n << " from " << source << " to " << destination <<
endl;
    total_moves++;

    // Move the (n-1) disks from auxiliary to destination
    towerOfHanoiMoves(n - 1, auxiliary, destination, source, total_moves);
}

int main() {
    int n;
    cin >> n;

    int total_moves = 0;
    towerOfHanoiMoves(n, 'A', 'C', 'B', total_moves);

    cout << "Total number of times the disk is moved: " << total_moves << endl;

    return 0;
}

```