

Consider a game where a player can score 3 or 5 or 10 points in a move. Given a total score n , find number of ways to reach the given score.

Examples:

Input: $n = 20$

Output: 4

There are following 4 ways to reach 20

(10, 10)

(5, 5, 10)

(5, 5, 5, 5)

(3, 3, 3, 3, 3, 5)

Input: $n = 13$

Output: 2

There are following 2 ways to reach 13

(3, 5, 5)

(3, 10)

We strongly recommend you to minimize the browser and try this yourself first.

This problem is a variation of [coin change problem](#) and can be solved in $O(n)$ time and $O(n)$ auxiliary space.

The idea is to create a table of size $n+1$ to store counts of all scores from 0 to n . For every possible move (3, 5 and 10), increment values in table.

```
// A C program to count number of possible ways to a given score
// can be reached in a game where a move can earn 3 or 5 or 10
#include <stdio.h>
```

```
// Returns number of ways to reach score n
int count(int n)
{
    // table[i] will store count of solutions for
    // value i.
    int table[n+1], i;

    // Initialize all table values as 0
    memset(table, 0, sizeof(table));

    // Base case (If given value is 0)
    table[0] = 1;

    // One by one consider given 3 moves and update the table[]
    // values after the index greater than or equal to the
    // value of the picked move
    for (i=3; i<=n; i++)
        table[i] += table[i-3];
    for (i=5; i<=n; i++)
        table[i] += table[i-5];
    for (i=10; i<=n; i++)
        table[i] += table[i-10];

    return table[n];
}
```

```
// Driver program
int main(void)
{
    int n = 20;
    printf("Count for %d is %d\n", n, count(n));
}
```

```
    n = 13;  
    printf("Count for %d is %d", n, count(n));  
    return 0;  
}
```

Output:

```
Count for 20 is 4  
Count for 13 is 2
```

Exercise: How to count score when (10, 5, 5), (5, 5, 10) and (5, 10, 5) are considered as different sequences of moves. Similarly, (5, 3, 3), (3, 5, 3) and (3, 3, 5) are considered different.