The Handshake Problem

The handshake problem is very simple to explain.

Basically, if you have a room full of people, how many handshakes are needed for each person to have shaken everybody else's hand exactly once?

For small groups, the solution is quite simple and can be counted fairly quickly, but what about 20 people? Or 50? Or 1000? In this article, we will look at how to work out the answers to these questions methodically and create a formula that can be used for any number of people.

Small Groups

Let's start by looking at solutions for small groups of people.

The answer is obvious for a group of 2 people: only 1 handshake is needed.

For a group of 3 people, person 1 will shake the hands of person 2 and person 3.

This leaves person 2 and 3 to shake hands with each other for a total of 3 handshakes.

For groups larger than 3, we will require a methodical way of counting to ensure we don't miss out or repeat any handshakes, but the math is still fairly simple.

Groups of Four People

Suppose we have four people in a room, whom we shall call A, B, C and D. We can split this into separate steps to make counting easier.

- ***Person A shakes hands with each of the other people in turn— 3 handshakes.***
- ***Person B has now shaken hands with A but still needs to shake hands with C and D— 2 more handshakes.***
- ***Person C has now shaken hands with A and B but still needs to* shake *D's hand— 1 more handshake.***
- ***Person D has now shaken hands with everybody.***

Our total number of handshakes is therefore:

$$\mathbf{3 + 2 + 1 = 6.}$$

Larger Groups

If you look closely at our calculation for the group of four, you can see a pattern that we can use to continue to work out the number of handshakes needed for different −sized groups.

Suppose we have `n` people in a room.

The first person shakes hands with everybody in the room except for himself.

His total number of handshakes is, therefore, one lower than the total number of people.

The second person has now shaken hands with the first person but still needs to shake hands with everybody else.

The number of people left is, therefore, two lower than the total number of people in the room.

The third person has now shaken hands with the first and second people.

That means the remaining number of handshakes for him is three lower than the total number of people in the room.

This continues with each person having one less handshake to make until we get to the penultimate person, who only has to shake hands with the last person.

Using this logic, we get the numbers of handshakes shown in the table below.

The Number of Handshakes Required for Different Sized Groups

Number of People In In the Room.	**Number of Handshakes Required**
2	**1**
3	$\mathbf{3(2+1)}$
4	$\mathbf{6(3+2+1)}$
5	$\mathbf{10(4+3+2+1)}$
6	$\mathbf{15(5+4+3+2+1)}$
7	$\mathbf{21(6+5+4+3+2+1)}$
8	$\mathbf{28(7+6+5+4+3+2+1)}$

$\textit{Creating a Formula for the Handshake}$

$\textbf{\textit{Our method so far is great for fairly small groupings,}}$

$\textbf{\textit{but it will still take a while for larger groups.}}$

$\textbf{\textit{For this reason, we will create an algebraic formula}}$

$\textbf{\textit{to instantly calculate the number of handshakes required}}$

$\textbf{\textit{for any size group.}}$

$\textbf{\textit{The total number of handshakes of `n` people}} =$

$$\mathbf{1 + 2 + \cdots + (n - 2) + (n - 1) + n = \frac{n \times (n - 1)}{2}}$$

$\textit{Lets do a program in non} - \textit{recursive way}$

```
int handshake(int n)
{
    if (n == 0 || n == 1)
    {
        return 0;
    }
    return (n*(n-1))/2;
}
```

Now Lets do a program in Recursive Way

```cpp
int handshake(int n)
{
    if (n == 0 || n == 1)
    {
        return 0;
    }
    cout<<n <<endl;
    return handshake(n-1)+(n-1);
}
```

Lets take: $n = 1$

Push

$n = 1$	*Value of n*
0	*Return Value*
$handshake(n)$	*MAIN FUNC()*

Pop

$n = 1$	*Value of n*
0	*Return Value*
$handshake(n)$	*MAIN FUNC()*

$$\boldsymbol{Lets\ take: n = 2}$$

$$\boldsymbol{Push}$$

$n = 1$	$Value\ of\ n$
0	$Return\ Value$
$n = 2$	$Value\ of\ n$
$+(n - 1\ i.e. 2 - 1 = 1)$	$Return\ Value$
$handshake(n)$	$MAIN\ FUNC()$

$$\boldsymbol{Pop}$$

$n = 1$	$Value\ of\ n$
0	$Return\ Value$
$n = 2$	$Value\ of\ n$
$0 + 1 = 1$	$Return\ Value$
1	$MAIN\ FUNC()$

$$\boldsymbol{Lets\ take: n = 3}$$

$$\boldsymbol{Push}$$

$n = 1$	$Value\ of\ n$
0	$Return\ Value$
$n = 2$	$Value\ of\ n$
$+(n - 1\ i.e. 2 - 1 = 1)$	$Return\ Value$
$n = 3$	$Value\ of\ n$
$+(n - 1\ i.e. 3 - 1 = 2)$	$Return\ Value$
$handshake(n)$	$MAIN\ FUNC()$

$$\boldsymbol{Pop}$$

$n = 1$	$Value\ of\ n$
0	$Return\ Value$
$n = 2$	$Value\ of\ n$
$0 + 1 = 1$	$Return\ Value$
$n = 3$	$Value\ of\ n$
$1 + 2 = 3$	$Return\ Value$
$handshake(n)$	$MAIN\ FUNC()$

$\boldsymbol{Lets\ take: n = 4}$

$\boldsymbol{Push}$

$n = 1$	$Value\ of\ n$
0	$Return\ Value$
$n = 2$	$Value\ of\ n$
$+(n - 1\ i.e. 2 - 1 = 1)$	$Return\ Value$
$n = 3$	$Value\ of\ n$
$+(n - 1\ i.e. 3 - 1 = 2)$	$Return\ Value$
$n = 4$	$Value\ of\ n$
$+(n - 1\ i.e. 4 - 1 = 3)$	$Return\ Value$
$handshake(n)$	$MAIN\ FUNC()$

$\boldsymbol{Pop}$

$n = 1$	$Value\ of\ n$
0	$Return\ Value$
$n = 2$	$Value\ of\ n$
$0 + 1 = 1$	$Return\ Value$
$n = 3$	$Value\ of\ n$
$1 + 2 = 3$	$Return\ Value$
$n = 4$	$Value\ of\ n$
$3 + 3 = 6$	$Return\ Value$
$handshake(n)$	$MAIN\ FUNC()$

And it goes in this way .

… … …