

A – Double it

Your mission, should you decide to accept it, is to double integers. You will be given the solution in Java, C++, and C so that you can observe how input and output are to be handled in the contest. It is suggested that you also try sending the source to the printer, and request a clarification from the judges (or just say 'hi!'). If you can't solve this given the solution to look at, you are hopeless!

Input:

There may be multiple cases, one per line with possible white space preceding and trailing. The last line has a single 0. A case consists of one integer which is to be doubled.

Output:

For each case display the double of the integer, formatted exactly as in the output below with "Case", a single space, the number of the case, a colon and one space, etc. and no white space following on the line.

Sample Input:

```
4
 25
-15
0
```

Sample Output:

```
Case 1: Double of 4 is 8
Case 2: Double of 25 is 50
Case 3: Double of -15 is -30
```

```

#include <iostream>
using namespace std;

// Sample C++ solution shows how to do I/O in the contest
// Exercise is to double each input number until 0 found

int main() {
    // Note that there is no "prompt for input" cout

    int c=1;
    int x;
    cin >> x;

    while (x != 0) {
        cout << "Case" << c++ << ": Double of " << x << " is "
             << 2*x << endl;
        cin >> x;
    }

    return 0;
}

```

```

-----

#include<stdio.h>

// Sample C solution shows how to do I/O in the contest
// Exercise is to double each input number until 0 found

int main() {
    // Note that there is no "prompt for input" printf

    int c=1;
    int x;
    scanf("%d", &x);

    while (x != 0) {
        printf("Case %d: Double of %d is %d\n", c++, x, 2*x);
        scanf("%d", &x);
    }

    return 0;
}

```

```

import java.util.Scanner;

// Sample Java solution shows how to do I/O in the contest
// Exercise is to double each input number until 0 found

public class Dubble {

    public static void main(String[] args) {
        // Note that there is no "prompt for input" println

        int c=1;
        int x;
        Scanner keyboard = new Scanner(System.in);

        x = keyboard.nextInt();

        while (x != 0) {
            System.out.println("Case " + c++ + ": Double of " +
                               x + " is " + 2*x);
            x = keyboard.nextInt();
        }
    }
}

```

```

import os, sys

// Sample Python 3 solution shows how to do I/O in the contest
// Exercise is to double each input number until 0 found

def do_stuff():
    // Note that there is no "prompt for input" print
    f = sys.stdin
    c = 0
    lines = f.readlines()
    for line in lines:
        nums = line.split()
        for n in nums:
            c = c + 1
            i = int( n )
            if i != 0:
                print ( "Case %d: Double of %d is %d"
                        % ( c, i, i * 2 ) )

if __name__ == "__main__":
    do_stuff()

```

B – Summing Up

Your mission, should you decide to accept it, is to add together all the integers within a given range. This may not be as easy as it appears. Good luck!

Input:

There may be multiple cases, one per line. The last line has two zeroes (0 0). A case consists of two integers, a and b . There may be additional white space in the line.

Output:

For each case display the sum of all the integers strictly between a and b . Format exactly as shown with “Case”, one space, the number of the case, a colon and one space, and the answer, with no following white space.

Sample Input:

```
2 6
 3 4
0 0
```

Sample Output:

```
Case 1: 12
Case 2: 0
```

C – It's a Fact!

All you need to do is calculate the factorials of the input numbers!

Input:

There may be multiple cases, one per line. The last line has 0. A case consists of a single positive integer up to 20. There may be white space in the lines.

Output:

For each case display the factorial of the input number. Format exactly as in the sample with "Case", one space, the number of the case, a colon and one space, then the answer with no trailing white space.

Sample Input:

```
4
1
  0
```

Sample Output:

```
Case 1: 24
Case 2: 1
```