

# Tipos de Input y Output

MSc. (c) Jhosimar George Arias Figueroa  
[jariasf03@gmail.com](mailto:jariasf03@gmail.com)

# Tipos de Input y Output

- En un concurso tenemos diversos tipos de Input y Output.
- La dificultad de un problema alguna veces esta en el Input y Output.
- Veremos distintos casos de input y output a continuación.

# CASO 1

- Para este caso el input del problema puede indicarnos lo siguiente:
  - Leer algun número o cadena hasta “End Of File / EOF”
  - El ingreso consiste de una serie de lineas ó de una serie de casos de prueba.
- Es probable que el problema no te indique cuan largo será el input.

# Ejemplo Caso 1

## Input

The input will consist of a series of pairs of integers a and b, separated by a space, one pair of integers per line.

## Output

For each pair of input integers a and b you should output the sum of a and b in one line, and with one line of output for each line in input.

### Sample Input

1 5  
10 20

### Sample Output

6  
30

# C++

```
//Usando scanf/printf y EOF ( End-Of-File)
int main(){
    int a , b;
    while( scanf("%d %d" , &a , &b ) != EOF ){
        printf("%d\n" , a + b );
    }
    return 0;
}
```

```
//Usando scanf/printf
int main(){
    int a , b;
    while( scanf("%d %d" , &a , &b ) == 2 ){
        printf("%d\n" , a + b );
    }
    return 0;
}
```

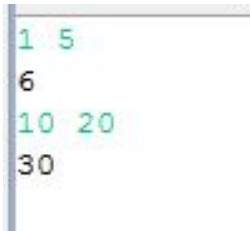
```
//Usando cin/cout
int main(){
    int a , b;
    while( cin>>a>>b ){
        cout<< a + b <<endl;
    }
    return 0;
}
```



```
1 5
6
10 20
30
-
```

# Java

```
//Usando Scanner
public static void main( String[] args ){
    Scanner sc = new Scanner( System.in );
    int a , b;
    while( sc.hasNext() ){
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println( a + b );
    }
}
```



```
1 5
6
10 20
30
```

```
//Usando BufferedReader y StringTokenizer
public static void main(String[] args) throws IOException {
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tokens;
    String next;
    int a , b;
    while( ( next = reader.readLine() ) != null ){
        tokens = new StringTokenizer( next );
        a = Integer.parseInt( tokens.nextToken() );
        b = Integer.parseInt( tokens.nextToken() );
        System.out.println( a + b );
    }
}
```

```
//BufferedReader y Split
public static void main(String[] args) throws IOException {
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    String next;
    int a , b;
    while( ( next = reader.readLine() ) != null ){
        String[] tokens = next.split("\\s+");
        a = Integer.parseInt( tokens[ 0 ] );
        b = Integer.parseInt( tokens[ 1 ] );
        System.out.println( a + b );
    }
}
```

# Ejemplo Caso 1

## Input

Input contains multiple test cases, and one case one line. Each case starts with an integer N, and then N integers follow in the same line.

## Output

For each test case you should output the sum of N integers in one line, and with one line of output for each line in input.

### Sample Input

4 1 2 3 4  
5 1 2 3 4 5

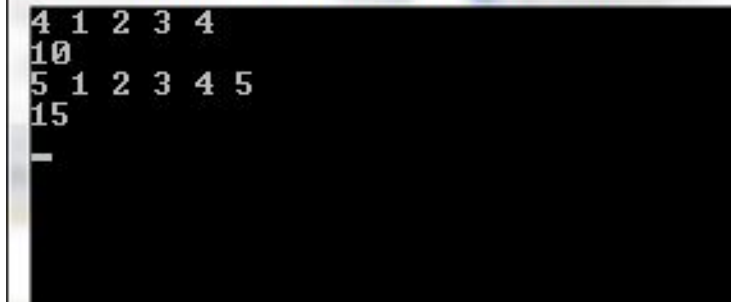
### Sample Output

10  
15

# C++

```
//Usando scanf/printf
int main() {
    int n , x , sum , i;
    while( scanf("%d" , &n ) != EOF ){
        sum = 0;
        for( i = 0 ; i < n ; ++i ){
            scanf("%d" , &x );
            sum += x;
        }
        printf("%d\n" , sum );
    }
    return 0;
}
```

```
//Usando cin/cout
int main() {
    int n , x , sum , i;
    while( cin>>n ){
        sum = 0;
        for( i = 0 ; i < n ; ++i ){
            cin>>x;
            sum += x;
        }
        cout<<sum<<endl;
    }
}
```

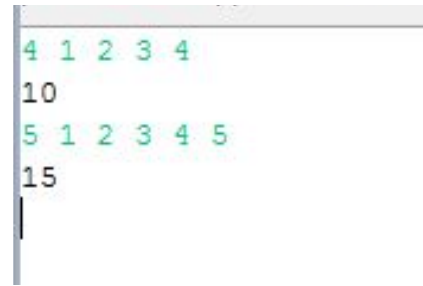


```
4 1 2 3 4
10
5 1 2 3 4 5
15
-
```



# Java

```
//Usando Scanner
public static void main(String[] args) {
    int n , x , sum , i;
    Scanner sc = new Scanner( System.in );
    while( sc.hasNext() ){
        n = sc.nextInt();
        sum = 0;
        for( i = 0 ; i < n ; ++i ){
            x = sc.nextInt();
            sum += x;
        }
        System.out.println( sum );
    }
}
```



```
4 1 2 3 4
10
5 1 2 3 4 5
15
```

```
//Usando BufferedReader y StringTokenizer
public static void main( String[] args ) throws IOException{
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tok = new StringTokenizer("");
    String next;
    int n , x , sum;
    while( ( next = reader.readLine() ) != null ){
        tok = new StringTokenizer( next );
        n = Integer.parseInt( tok.nextToken() );
        sum = 0;
        while( n-- > 0 ){
            x = Integer.parseInt( tok.nextToken() );
            sum += x;
        }
        System.out.println( sum );
    }
}
```

# CASO 2

- Para este caso el input del problema puede indicarnos lo siguiente:
  - El ingreso consiste de un entero  $N$  en la primera línea y entonces  $N$  líneas son ingresadas.
  - El ingreso consiste de un entero  $N$  que denota el número de casos de prueba, cada caso contiene...

# Ejemplo Caso 2

## Input

Input contains an integer N in the first line, and then N lines follow. Each line consists of a pair of integers a and b, separated by a space, one pair of integers per line.

## Output

For each pair of input integers a and b you should output the sum of a and b in one line, and with one line of output for each line in input.

### Sample Input

```
2
1 5
10 20
```

### Sample Output

```
6
30
```

# C++

```
//Usando scanf/printf y bucle while
int main(){
    int a , b , t;
    scanf("%d" , &t );
    while( t-- ){
        scanf("%d %d" , &a , &b );
        printf("%d\n" , a + b );
    }
    return 0;
}
```

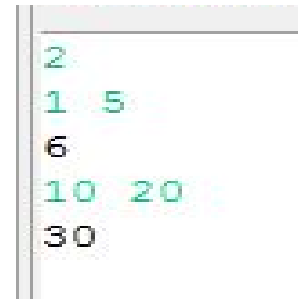
```
//Usando scanf/printf y bucle for
int main(){
    int a , b , t;
    scanf("%d" , &t );
    for( int q = 1 ; q <= t ; ++q ){
        scanf("%d %d" , &a , &b );
        printf("%d\n" , a + b );
    }
    return 0;
}
```

```
//Usando cin/cout
int main(){
    int a , b , t;
    cin>>t;
    for( int q = 1 ; q <= t ; ++q ){
        cin>>a>>b;
        cout<< a + b <<endl;
    }
    return 0;
}
```



# Java

```
//Usando Scanner
public static void main(String[] args) {
    Scanner sc = new Scanner( System.in );
    int n = sc.nextInt() , a , b;
    while( n-- > 0 ){
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println( a + b );
    }
}
```



```
2
1 5
6
10 20
30
```

```
//Usando BufferedReader y StringTokenizer
public static void main(String[] args) throws IOException {
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tokens;
    String next = reader.readLine();
    tokens = new StringTokenizer( next );
    int n = Integer.parseInt( tokens.nextToken() ) , a , b;
    while( n-- > 0 ){
        next = reader.readLine();
        tokens = new StringTokenizer( next );
        a = Integer.parseInt( tokens.nextToken() );
        b = Integer.parseInt( tokens.nextToken() );
        System.out.println( a + b );
    }
}
```

# Caso 3

- Este caso puede iniciar de manera similar que el caso 1:
  - El ingreso contiene múltiples o varios casos de prueba.
- Pero en la parte final del input tendremos lo siguiente:
  - El último caso es seguido por: n ceros | una cadena | un mensaje.
  - Un caso conteniendo - n ceros | una cadena | un mensaje - termina el input, este caso no será procesado.

# Ejemplo Caso 3

## Input

Input contains multiple test cases. Each test case contains a pair of integers a and b, one pair of integers per line. A test case containing 0 0 terminates the input and this test case is not to be processed.

## Output

For each pair of input integers a and b you should output the sum of a and b in one line, and with one line of output for each line in input.

### Sample Input

```
1 5
10 20
0 0
```

### Sample Output

```
6
30
```

# C++

//Usando scanf/printf y condicional if

```
int main(){
    int a , b;
    while( scanf("%d %d" , &a , &b ) == 2 ){
        if( a == 0 && b == 0 ) break;
        printf("%d\n" , a + b );
    }
    return 0;
}
```

//Usando scanf/printf

```
int main(){
    int a , b;
    while( scanf("%d %d" , &a , &b ) && a|b ){
        printf("%d\n" , a + b );
    }
    return 0;
}
```

//Usando cin/cout

```
int main(){
    int a , b;
    while( cin>>a>>b , a + b ){
        cout<< a + b <<endl;;
    }
    return 0;
}
```

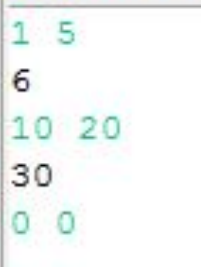


```
1 5
6
10 20
30
0 0
```



# Java

```
//Usando Scanner
public static void main(String[] args) {
    Scanner sc = new Scanner( System.in );
    int a , b;
    while( true ){
        a = sc.nextInt();
        b = sc.nextInt();
        if( a == 0 && b == 0 ) break;
        System.out.println( a + b );
    }
}
```



A terminal window showing the output of the Scanner program. The input sequence is 1 5, 6, 10 20, 30, and 0 0. The output shows the sum of each pair of numbers: 6, 16, 30, and 0. The input is shown in green text, and the output is shown in red text.

```
1 5
6
10 20
30
0 0
6
16
30
0
```

```
//Usando BufferedReader y StringTokenizer
public static void main( String[] args ) throws IOException{
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tokens;
    int a , b;
    while( true ){
        tokens = new StringTokenizer( reader.readLine() );
        a = Integer.parseInt( tokens.nextToken() );
        b = Integer.parseInt( tokens.nextToken() );
        if( ( a | b ) == 0 ) break;
        System.out.printf( "%d%n" , a + b );
    }
}
```

# Ejemplo Caso 3

## Input

Input contains multiple test cases. Each test case contains a integer N, and then N integers follow in the same line. A test case starting with 0 terminates the input and this test case is not to be processed.

## Output

For each group of input integers you should output their sum in one line, and with one line of output for each line in input.

### Sample Input

```
4 1 2 3 4
5 1 2 3 4 5
0
```

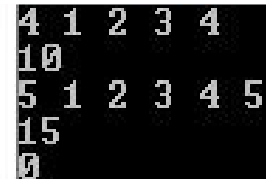
### Sample Output

```
10
15
```

# C++

```
//Usando scanf/printf
int main(){
    int n , a , sum;
    while( scanf("%d" , &n ) , n ){
        sum = 0;
        while( n-- ){
            scanf("%d" , &a );
            sum += a;
        }
        printf("%d\n" , sum );
    }
    return 0;
}
```

```
//Usando cin/cout
int main(){
    int n , a , sum;
    while( cin>>n , n ){
        sum = 0;
        while( n-- ){
            cin>>a;
            sum += a;
        }
        cout<<sum<<endl;
    }
}
```



```
4 1 2 3 4
10
5 1 2 3 4 5
15
0
```

# Java

```
//Usando Scanner
public static void main(String[] args) {
    int n , x , sum , i;
    Scanner sc = new Scanner( System.in );
    while( ( n = sc.nextInt() ) != 0 ){
        sum = 0;
        for( i = 0 ; i < n ; ++i ){
            x = sc.nextInt();
            sum += x;
        }
        System.out.println( sum );
    }
}
```

```
5 1 2 3 4 5
15
4 1 2 3 4
10
0
```

```
//Usando BufferedReader y StringTokenizer
public static void main( String[] args ) throws IOException{
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tok = new StringTokenizer("");
    int n , x , sum;
    while( true ){
        tok = new StringTokenizer( reader.readLine() );
        n = Integer.parseInt( tok.nextToken() );
        if( n == 0 ) break;
        sum = 0;
        while( n-- > 0 ){
            x = Integer.parseInt( tok.nextToken() );
            sum += x;
        }
        System.out.println( sum );
    }
}
```

# Caso 4

- Para este caso el output puede indicarnos lo siguiente:
  - La salida de 2 casos consecutivos será separado por una línea en blanco.
  - Imprime una línea en blanco entre casos.

Salida para el input 1 //Como se describe en el enunciado del problema

--Linea en Blanco---

Salida para el input 2 //Como se describe en el enunciado del problema

-- Linea en Blanco ---

Salida para el input 3 //Como se describe en el enunciado del problema

-- Linea en Blanco ---

.

.

.

-- Linea en Blanco ---

Salida para el input n //Como se describe en el enunciado del problema

--Fin de Archivo--

# Ejemplo Caso 4

## Input

Input contains an integer N in the first line, and then N lines follow. Each line starts with a integer M, and then M integers follow in the same line.

## Output

For each group of input integers you should output their sum in one line, and you must note that there is a blank line between outputs.

### Sample Input

```
3
4 1 2 3 4
5 1 2 3 4 5
3 1 2 3
```

### Sample Output

```
10

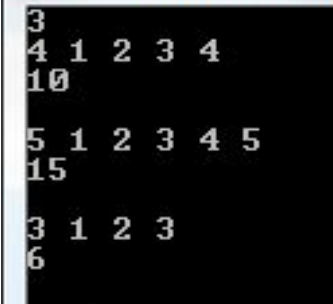
15

6
```

# C++

```
//Usando scanf/printf
int main(){
    int n , a , sum , t;
    scanf("%d" , &t );
    for( int q = 0 ; q < t ; ++q ){
        if( q ) printf("\n");
        scanf("%d" , &n );
        sum = 0;
        for( int i = 0 ; i < n ; ++i ){
            scanf("%d" , &a );
            sum += a;
        }
        printf("%d\n" , sum );
    }
    return 0;
}
```

```
//Usando cin/cout
int main(){
    int n , a , sum , t;
    cin>>t;
    while( t-- ){
        cin>>n;
        sum = 0;
        for( int i = 0 ; i < n ; ++i ){
            cin>>a;
            sum += a;
        }
        cout<<sum<<endl;
        if( t ) cout<<endl;
    }
    return 0;
}
```

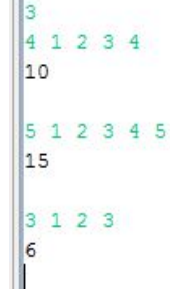


```
3
4 1 2 3 4
10
5 1 2 3 4 5
15
3 1 2 3
6
```

# Java

```
//Usando Scanner
public static void main( String[] args ){
    Scanner sc = new Scanner( System.in );
    int x , t , n , sum;
    t = sc.nextInt();
    while( t-- > 0 ){
        n = sc.nextInt();
        sum = 0;
        while( n-- > 0 ){
            x = sc.nextInt();
            sum += x;
        }
        System.out.println( sum );
        if( t != 0 ){
            System.out.println();
        }
    }
}
```

```
//Usando BufferedReader y StringTokenizer
public static void main( String[] args ) throws IOException{
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tokens;
    int n , x , sum , t;
    t = Integer.parseInt( reader.readLine() );
    for( int q = 0 ; q < t ; ++q ){
        if( q > 0 ) System.out.println();
        tokens = new StringTokenizer( reader.readLine() );
        n = Integer.parseInt( tokens.nextToken() );
        sum = 0;
        while( n-- > 0 ){
            x = Integer.parseInt( tokens.nextToken() );
            sum += x;
        }
        System.out.println( sum );
    }
}
```



```
3
4 1 2 3 4
10

5 1 2 3 4 5
15

3 1 2 3
6
```



# Caso 5

- Para este caso el output puede indicarnos lo siguiente:
  - Una línea de blanco deberá seguir cada grupo de salida.
  - Después de cada caso usted deberá imprimir una línea en blanco.
  - Imprime una linea en blanco después de cada caso(incluso después del último caso).

Salida para el input 1 //Como se describe en el enunciado del problema

--Linea en Blanco---

Salida para el input 2 //Como se describe en el enunciado del problema

-- Linea en Blanco ---

Salida para el input 3 //Como se describe en el enunciado del problema

-- Linea en Blanco ---

.

.

-- Linea en Blanco ---

Salida para el input n //Como se describe en el enunciado del problema

-- Linea en Blanco ---

--Fin de Archivo--

# Ejemplo Caso 5

## Input

The input will consist of a series of pairs of integers a and b, separated by a space, one pair of integers per line.

## Output

For each pair of input integers a and b you should output the sum of a and b, and followed by a blank line.

### Sample Input

```
1 5
10 20
```

### Sample Output

```
6

30
```

# C++

```
//Usando scanf/printf
int main(){
    int a , b;
    while( scanf("%d %d" , &a , &b ) != EOF ){
        printf("%d\n\n" , a + b );
    }
    return 0;
}
```

```
//Usando cin/cout
int main(){
    int a , b;
    while( cin>>a>>b ){
        cout<< a + b <<endl<<endl;;
    }
    return 0;
}
```

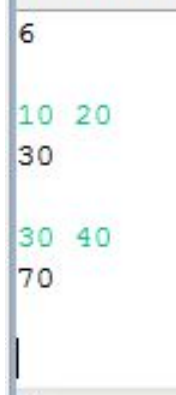


A terminal window with a black background and white text. It shows the output of a C++ program using scanf/printf. The first line shows '1 5' followed by a newline, then '6' followed by two newlines. The second line shows '10 20' followed by a newline, then '30' followed by two newlines. A horizontal line is visible at the bottom of the terminal.

```
1 5
6
10 20
30
-
```

# Java

```
//Usando Scanner y Println
public static void main( String[] args ){
    Scanner sc = new Scanner( System.in );
    int a , b;
    while( sc.hasNext() ){
        a = sc.nextInt();
        b = sc.nextInt();
        System.out.println( a + b );
        System.out.println();
    }
}
```



A screenshot of a Java program's output. It shows a series of numbers printed on separate lines: 6, 10 20, 30, 30 40, and 70. The numbers 10 20 and 30 40 are printed in green, while the others are in black. The output is displayed in a window with a light blue border.

```
//Usando BufferedReader y Printf
public static void main(String[] args) throws IOException {
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tokens;
    String next;
    int a , b;
    while( ( next = reader.readLine() ) != null ){
        tokens = new StringTokenizer( next );
        a = Integer.parseInt( tokens.nextToken() );
        b = Integer.parseInt( tokens.nextToken() );
        System.out.printf("%d%n%n" , a + b );
    }
}
```

# Caso 6

- Este caso me indicará lo siguiente en el output:
  - Para cada caso, imprime – n números | n cadenas
  - separados por un espacio.

Salida para input 1: Salida1-*Espacio*-Salida2-*Espacio*-...-*Espacio*-SalidaN

Salida para input 2: Salida1-*Espacio*-Salida2-*Espacio*-...-*Espacio*-SalidaN

.

.

.

Salida para input n: Salida1-*Espacio*-Salida2-*Espacio*-...-*Espacio*-SalidaN

--Fin de Archivo--

# Ejemplo Caso 6

## Input

Input contains multiple test cases. The first line of the input is a single integer  $T$  which is the number of test cases.  $T$  test cases follow. Each test case contains an integer  $N$  ( $1 \leq N \leq 1000$  the number of integers to be sorted) and then  $N$  integers follow in the same line. It is guaranteed that all integers are in the range of 32-bit.

## Output

For each case, print the sorting result, and one line one case.

### Sample Input

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

### Sample Output

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

# C++

//Impresión de primer elemento y siguientes en bucle

```
#define MAX 1005
```

```
int main(){
```

```
    int t , n , a[ MAX ];
```

```
    scanf("%d" , &t );
```

```
    while( t-- ){
```

```
        scanf("%d" , &n );
```

```
        for( int i = 0 ; i < n ; ++i ) scanf("%d" , &a[ i ]);
```

```
        sort( a , a + n );
```

```
        printf("%d" , a[ 0 ] );
```

```
        for( int i = 1 ; i < n ; ++i ) printf(" %d" , a[ i ]);
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

//Impresión de todos los elementos en bucle

```
#define MAX 1005
```

```
int main(){
```

```
    int t , n , a[ MAX ];
```

```
    scanf("%d" , &t );
```

```
    while( t-- ){
```

```
        scanf("%d" , &n );
```

```
        for( int i = 0 ; i < n ; ++i ) scanf("%d" , &a[ i ]);
```

```
        sort( a , a + n );
```

```
        for( int i = 0 ; i < n ; ++i ){
```

```
            if( i ) printf(" ");
```

```
            printf("%d" , a[ i ]);
```

```
        }
```

```
        printf("\n");
```

```
    }
```

```
    return 0;
```

```
}
```

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

# Caso 7

- Este caso puede indicarnos lo siguiente en el input:
  - El ingreso consiste de varias líneas, cada línea contiene varias palabras ó números separados por uno o más espacios.
  - Cada ingreso es separado por una línea en blanco ó el ingreso termina con una línea en blanco.
- Tener cuidado de combinar:
  - C++: scanf – gets, cin – getline.
  - Java: nextInt – nextLine.



## Input

The first line of input gives the number of cases,  $T$  ( $1 \leq T \leq 30$ ). Each test case represents a message, which is composed by  $1 \leq N \leq 100$  lines and each line is composed by  $1 \leq M \leq 30$  words. Two words in the same line are separated by one or more white spaces. There will be a blank line after each message.

## Output

For each test case you must print the number of the test case and each word of message, one per line (look the sample output for the exact format). You must print a blank line between each test case.

### Sample Input

```
2
Hey    good
    lawyer  as

First I give
money Teresa
```

### Sample Output

```
Case #1:
Hey
good
lawyer
as

Case #2:
First
I
give
money
Teresa
```

# C++

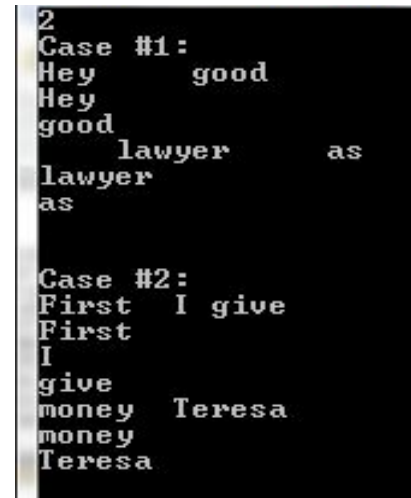
//Usando librerias de C puro

```
int main(){
    char line[ 105 ];
    int t , len;
    char *token;
    scanf("%d" , &t );
    gets( line );
    for( int q = 0 ; q < t ; ++q ){
        if( q ) printf("\n");
        printf("Case #%d:\n" , q + 1 );
        while( gets( line ) ){
            len = strlen( line );
            if( len == 0 ) break;
            token = strtok( line , " " );
            while( token != NULL ){
                printf("%s\n" , token );
                token = strtok( NULL , " " );
            }
        }
    }
    return 0;
}
```

//Usado librerias de C++

```
int main(){
    int t;
    string line , token;
    cin>>t;
    getline( cin , line );
    for( int q = 1 ; q <= t ; ++q ){
        cout<<"Case #"<<q<<":\n";
        while( getline( cin , line ) , line != "" ){
            stringstream ss( line );
            while( ss>>token ){
                cout<<token<<endl;
            }
        }
        if( q != t ) cout<<endl;
    }
    return 0;
}
```

```
gets( line );
sscanf( line , "%d" , &t );
for( int q = 0 ; q < t ; ++q ){
    ...
}
```



```
2
Case #1:
Hey
good
Hey
good
lawyer
as
lawyer
as

Case #2:
First
I
give
money
Teresa
money
Teresa
```

# JAVA

```
//Usando Scanner y split
public static void main(String[] args) {
    Scanner sc = new Scanner( System.in );
    String tokens[];
    int t = sc.nextInt();
    String line = sc.nextLine();
    for( int q = 0 ; q < t ; ++q ){
        if( q > 0 ) System.out.println();
        System.out.println("Case #" + ( q + 1 ) + ":");
        while( !( line = sc.nextLine() ).equals("") ){
            tokens = line.trim().split("\\s+");
            for( int i = 0 ; i < tokens.length ; ++i ){
                System.out.println( tokens[ i ] );
            }
        }
    }
}

//Usando BufferedReader y StringTokenizer
public static void main(String[] args) throws IOException {
    BufferedReader reader = new BufferedReader( new InputStreamReader( System.in ) );
    StringTokenizer tokens;
    String line;
    int t = Integer.parseInt( reader.readLine() );
    for( int q = 1 ; q <= t ; ++q ){
        System.out.printf("Case %d:%n" , q );
        while( !( line = reader.readLine() ).equals( "" ) ){
            tokens = new StringTokenizer( line , " " );
            while( tokens.hasMoreTokens() ){
                System.out.printf( "%s%n" , tokens.nextToken() );
            }
        }
        if( q != t ) System.out.printf("%n" , q );
    }
}
```

```
2
Case #1:
Hey      good
Hey
good
        lawyer      as
lawyer
as

Case #2:
First   I give
First
I
give
money  Teresa
money
Teresa
```

# Ejercicios

- [A+B for Input-Output Practice \(I\)](#)
- [A+B for Input-Output Practice \(II\)](#)
- [A+B for Input-Output Practice \(III\)](#)
- [A+B for Input-Output Practice \(IV\)](#)
- [A+B for Input-Output Practice \(V\)](#)
- [A+B for Input-Output Practice \(VI\)](#)
- [A+B for Input-Output Practice \(VII\)](#)
- [A+B for Input-Output Practice \(VIII\)](#)
- [As Easy as A+B](#)
- [Decoding the message](#)