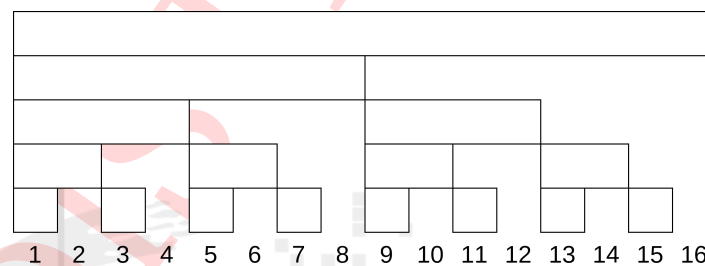


## Ancient Fractal

Jojo found a fractal art made by ancient civilization, which can be seen in the illustration below. In order to preserve the art, he decided to make an algorithm to generate the fractal. During his research, here is what he found about the fractal. The fractal consists of infinitely many horizontal segments, which are numbered by integers started from 1. The  $i$ -th segment is represented by the range  $[X_L[i], X_R[i]]$ , where  $X_L[i]$  and  $X_R[i]$  are integers, and  $X_L[i] \leq X_R[i]$ .



Here is the algorithm Jojo invented to construct the fractal. The fractal starts with the first segment  $X_L[1] = X_R[1] = 1$ . Then, for every iteration, it follows these steps:

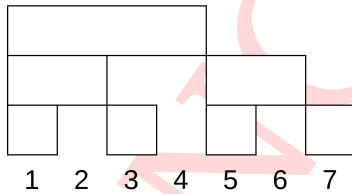
- Supposed that segment 1 to  $K$  have been generated by the previous iteration.
- First, add  $(K + 1)$ -th segment where  $X_L[K + 1] = 1$  and  $X_R[K + 1] = K + 1$ .
- Then, the next  $K$  segments are the translation of the first  $K$  segments. Formally, for all segment  $i$  where  $1 \leq i \leq K$ , add  $(K + 1 + i)$ -th segment where  $X_L[K + 1 + i] = X_L[i] + K$  and  $X_R[K + 1 + i] = X_R[i] + K$ .

At first, there is only segment 1. After the first iteration, there will be  $1 + 1 + 1 = 3$  segments. After the second iteration, there will be  $3 + 1 + 3 = 7$  segments. After the third iteration, there will be  $7 + 1 + 7 = 15$  segments. By repeating the iteration infinitely many times, the whole fractal can be created.

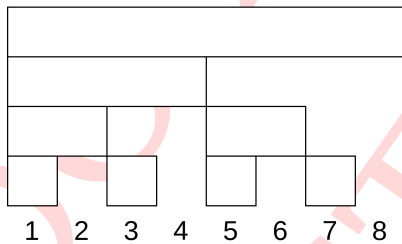
Unfortunately, Jojo struggles with the implementation. He asked you to create a program which can output the fractal art. Since it is impossible to create a program which output infinitely many segments, Jojo only asked you to output  $X_L[i]$  and  $X_R[i]$  for all segments  $i$  where  $A \leq i \leq B$ .

The following illustration is the step by step of creating the segments from the second iteration to the third iteration.

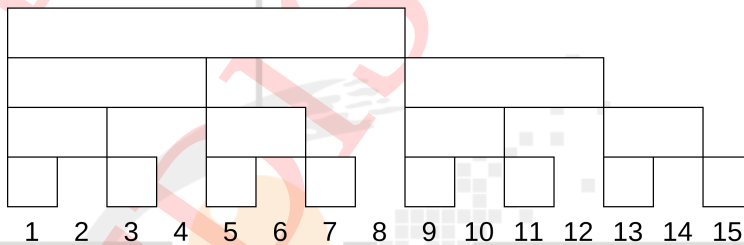
1. Initial Condition from the 2nd iteration. There are 7 segments.



2. Add the 8th segment, from X=1 to 8



3. Translate the 1st..7th segments to 9th..15th, and this is the result of the 3rd iteration.



## Format Input

A single line consists of 2 integers,  $A$  and  $B$ .

## Format Output

Output  $(B - A + 1)$  lines. Each line consists of 2 integers, which describes the asked segments.

## Constraints

- $1 \leq A \leq B \leq 10^9$
- $1 \leq (B - A + 1) \leq 3 \times 10^4$

## Sample Input 1 (standard input)

```
1 16
```

### Sample Output 1 (standard output)

```
1 1
1 2
3 3
1 4
5 5
5 6
7 7
1 8
9 9
9 10
11 11
9 12
13 13
13 14
15 15
1 16
```

### Sample Input 2 (standard input)

```
999999994 1000000000
```

### Sample Output 2 (standard output)

```
999999993 999999994
999999995 999999995
999999993 999999996
999999997 999999997
999999997 999999998
999999999 999999999
999999489 1000000000
```

### Explanation

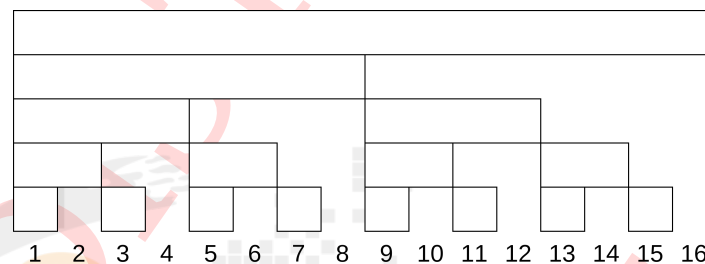
The first illustration represents the output in the first sample.

### Hint

It is possible to solve this problem using recursion. It is also possible to solve this problem using only looping and bitwise manipulation.

## Ancient Fractal

Jojo menemukan lukisan fraktal yang dibuat oleh peradaban kuno, yang dapat Anda lihat pada ilustrasi di bawah. Untuk melindungi lukisan tersebut, ia memutuskan untuk membuat sebuah algoritma yang dapat membuat fraktal tersebut. Selama riset, berikut ini adalah apa yang Jojo temukan. Fraktal tersebut tersusun atas segmen-segmen dengan jumlah tak hingga, yang dinomori dengan bilangan bulat mulai dari 1. Segmen ke- $i$  direpresentasikan dengan sebuah jangkauan  $[X_L[i], X_R[i]]$ , dimana  $X_L[i]$  dan  $X_R[i]$  adalah bilangan bulat, dan  $X_L[i] \leq X_R[i]$ .



Berikut adalah algoritma yang Jojo temukan untuk membuat fraktal tersebut. Fraktal dimulai dengan segment pertama  $X_L[1] = X_R[1] = 1$ . Kemudian, untuk setiap iterasi, akan dilakukan langkah-langkah berikut:

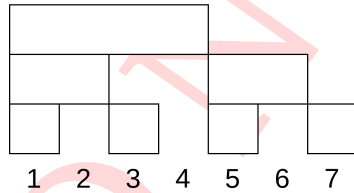
- Misalkan segmen ke-1 hingga segmen ke- $K$  sudah dibentuk pada iterasi sebelumnya.
- Pertama, tambahkan segmen ke- $(K + 1)$  dimana  $X_L[K + 1] = 1$  dan  $X_R[K + 1] = K + 1$ .
- Kemudian,  $K$  segment berikutnya adalah translasi dari  $K$  segment pertama. Secara formal, untuk semua segmen  $i$  dimana  $1 \leq i \leq K$ , tambahkan segmen ke- $(K + 1 + i)$  dimana  $X_L[K + 1 + i] = X_L[i] + K$  dan  $X_R[K + 1 + i] = X_R[i] + K$ .

Awalnya, hanya terdapat segmen 1. Setelah iterasi pertama, akan terdapat  $1 + 1 + 1 = 3$  segmen. Setelah iterasi kedua, akan terdapat  $3 + 1 + 3 = 7$  segmen. Setelah iterasi ketiga, akan terdapat  $7 + 1 + 7 = 15$  segmen. Dengan mengulangi iterasi tersebut sejumlah tak hingga, keseluruhan fraktal dapat dibuat.

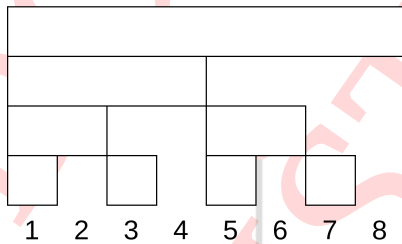
Sayangnya, Jojo kesulitan dengan implementasi nya. Ia meminta tolong Anda untuk membuat program yang dapat mengeluarkan lukisan fraktal tersebut. Karena tidak mungkin untuk membuat program yang mengeluarkan segmen dengan jumlah tak hingga, Jojo hanya meminta Anda untuk mengeluarkan  $X_L[i]$  dan  $X_R[i]$  untuk semua segmen  $i$  dimana  $A \leq i \leq B$ .

Ilustrasi berikut adalah tahapan membuat segmen-segmen dari iterasi ke-2 hingga iterasi ke-3.

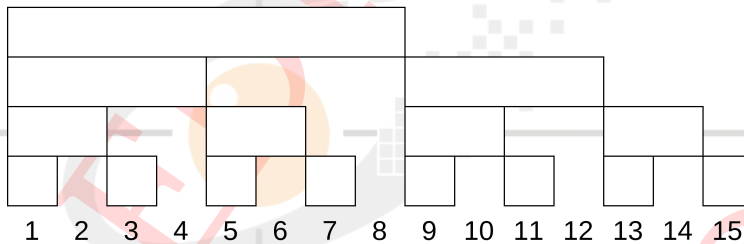
1. Initial Condition from the 2nd iteration. There are 7 segments.



2. Add the 8th segment, from X=1 to 8



3. Translate the 1st..7th segments to 9th..15th, and this is the result of the 3rd iteration.



## Format Input

Sebuah baris berisi 2 bilangan bulat,  $A$  dan  $B$ .

## Format Output

Keluarkan  $(B - A + 1)$  baris. Setiap baris tersusun atas 2 bilangan bulat, yang menggambarkan segmen yang diminta.

## Constraints

- $1 \leq A \leq B \leq 10^9$
- $1 \leq (B - A + 1) \leq 3 \times 10^4$

## Sample Input 1 (standard input)

```
1 16
```

### Sample Output 1 (standard output)

```
1 1
1 2
3 3
1 4
5 5
5 6
7 7
1 8
9 9
9 10
11 11
9 12
13 13
13 14
15 15
1 16
```

### Sample Input 2 (standard input)

```
999999994 1000000000
```

### Sample Output 2 (standard output)

```
999999993 999999994
999999995 999999995
999999993 999999996
999999997 999999997
999999997 999999998
999999999 999999999
999999489 1000000000
```

### Explanation

Ilustrasi di atas merepresentasikan keluaran yang ada di contoh kasus uji pertama.

### Hint

Soal ini dapat diselesaikan menggunakan rekursi. Soal ini juga dapat diselesaikan hanya dengan perulangan dan manipulasi *bitwise*.