A pointer in C is a way to share a memory address among different contexts (primarily functions). They are primarily used whenever a function needs to modify the content of a variable, of which it doesn't have ownership.

In order to access the memory address of a variable, val, we need to prepend it with $\&$ sign. E.g., `&val` returns the memory address of val.

This memory address is assigned to a pointer and can be shared among various functions. E.g. $int*p = \&val$ will assign the memory address of val to pointer p. To access the content of the memory to which the pointer points, prepend it with a `*`. For example, `*p` will return the value reflected by val and any modification to it will be reflected at the source (val).

```
void increment(int *v) {
    (*v)++;
}

int main() {
    int a;
    scanf("%d", &a);
    increment(&a);
    printf("%d", a);
    return 0;
}
```

You have to complete the function *void update(int *a,int *b)*, which reads two integers as argument, and sets a with the sum of them, and b with the absolute difference of them.

- $a' = a + b$
- $b' = |a - b|$

Input Format

Input will contain two integers, a and b, separated by a newline.

Output Format

You have to print the updated value of a and b, on two different lines.

P.S.: Input/ouput will be automatically handled. You only have to complete the *void update(int *a,int *b)* function.

Sample Input

```
4
5
```

Sample Output

```
9
1
```

Explanation

- $a' = 4 + 5 = 9$
- $b' = |4 - 5| = 1$