Structs

struct is a way to combine multiple fields to represent a composite data structure, which further lays the foundation for Object Oriented Programming. For example, we can store details related to a student in a struct consisting of his *age (int), first_name (string), last_name (string) and standard (int)*.

struct can be represented as

```
struct NewType {
    type1 value1;
    type2 value2;
    .
    .
    .
    typeN valueN;
};
```

You have to create a struct, named *Student*, representing the student's details, as mentioned above, and store the data of a student.

Input Format

Input will consist of four lines.
The first line will contain an integer, representing *age*.
The second line will contain a string, consisting of lower-case Latin characters ('a'-'z'), representing the *first_name* of a student.
The third line will contain another string, consisting of lower-case Latin characters ('a'-'z'), representing the *last_name* of a student.
The fourth line will contain an integer, representing the *standard* of student.

Note: The number of characters in *first_name* and *last_name* will not exceed 50.

Output Format

Output will be of a single line, consisting of *age*, *first_name*, *last_name* and *standard*, each separated by one white space.

P.S.: I/O will be handled by HackerRank.

Sample Input

```
15
john
carmack
10
```

Sample Output

```
15 john carmack 10
```