Basic Data Types

Some *C++* data types, their format specifiers, and their most common bit widths are as follows:

- *Int ("%d"):* 32 Bit integer
- *Long ("%ld"):* 32 bit integer (same as Int for modern systems)
- *Long Long ("%lld"):* 64 bit integer
- *Char ("%c"):* Character type
- *Float ("%f"):* 32 bit real value
- *Double ("%lf"):* 64 bit real value

Reading
To read a data type, use the following syntax:

```
scanf("`format_specifier`", &val)
```

For example, to read a *character* followed by a *double*:

```
char ch;
double d;
scanf("%c %lf", &ch, &d);
```

For the moment, we can ignore the spacing between format specifiers.

Printing
To print a data type, use the following syntax:

```
printf("`format_specifier`", val)
```

For example, to print a *character* followed by a *double*:

```
char ch = 'd';
double d = 234.432;
printf("%c %lf", ch, d);
```

Note: You can also use *cin* and *cout* instead of *scanf* and *printf*; however, if you are taking a million numbers as input and printing a million lines, it is faster to use *scanf* and *printf*.

Input Format

Input consists of the following space-separated values: *int*, *long*, *long*, *long*, *char*, *float*, and *double*, respectively.

Output Format

Print each element on a new line in the same order it was received as input. Note that the floating point value should be correct up to 3 decimal places and the double to 9 decimal places.

Sample Input

```
3 444 12345678912345 a 334.23 14049.30493
```

Sample Output

```
3
444
12345678912345
a
334.230
14049.304930000
```

Explanation

Print *int* 3,
followed by *long* 444,
followed by *long long* 12345678912345,
followed by *char* a,
followed by *float* 334.23,
followed by *double* 14049.30493.