Conditional Statements

if and *else* are two of the most frequently used conditionals in C/C++, and they enable you to execute zero or one conditional statement among many such dependent conditional statements. We use them in the following ways:

1. *if:* This executes the body of bracketed code starting with $\textit{statement1}$ if $\textit{condition}$ evaluates to *true*.

```
if (condition) {
   statement1;
   ...
}
```

2. *if - else:* This executes the body of bracketed code starting with $\textit{statement1}$ if $\textit{condition}$ evaluates to *true*, or it executes the body of code starting with $\textit{statement2}$ if $\textit{condition}$ evaluates to *false*. Note that only *one* of the bracketed code sections will ever be executed.

```
if (condition) {
   statement1;
   ...
}
else {
   statement2;
   ...
}
```

3. *if - else if - else:* In this structure, dependent statements are chained together and the $\textit{condition}$ for each statement is only checked if all prior conditions in the chain evaluated to *false*. Once a $\textit{condition}$ evaluates to *true*, the bracketed code associated with that statement is executed and the program then skips to the end of the chain of statements and continues executing. If each $\textit{condition}$ in the chain evaluates to false, then the body of bracketed code in the *else* block at the end is executed.

```
if(first condition) {
   ...
}
else if(second condition) {
   ...
}
.
.
.
else if((n-1)'th condition) {
   ....
}
else {
   ...
}
```

Given a positive integer denoting n, do the following:

- If $1 \leq n \leq 9$, then print the lowercase English word corresponding to the number (e.g., `one` for 1, `two` for 2, etc.).
- If $n > 9$, print `Greater than 9`.

Input Format

A single integer denoting n.

Constraints

- $1 \leq n \leq 10^9$

Output Format

If $1 \leq n \leq 9$, then print the lowercase English word corresponding to the number (e.g., `one` for 1, `two` for 2, etc.); otherwise, print `Greater than 9` instead.

Sample Input 0

```
5
```

Sample Output 0

```
five
```

Explanation 0

`five` is the English word for the number 5.

Sample Input 1

```
8
```

Sample Output 1

```
eight
```

Explanation 1

`eight` is the English word for the number 8.

Sample Input 2

```
44
```

Sample Output 2

```
Greater than 9
```

Explanation 2

$n = 44$ is greater than 9, so we print `Greater than 9`.