Input Format

A single line of text denoting $inputString$ (the variable whose contents must be printed).

Output Format

Print `Hello, World.` on the first line, and the contents of $inputString$ on the second line.

Sample Input

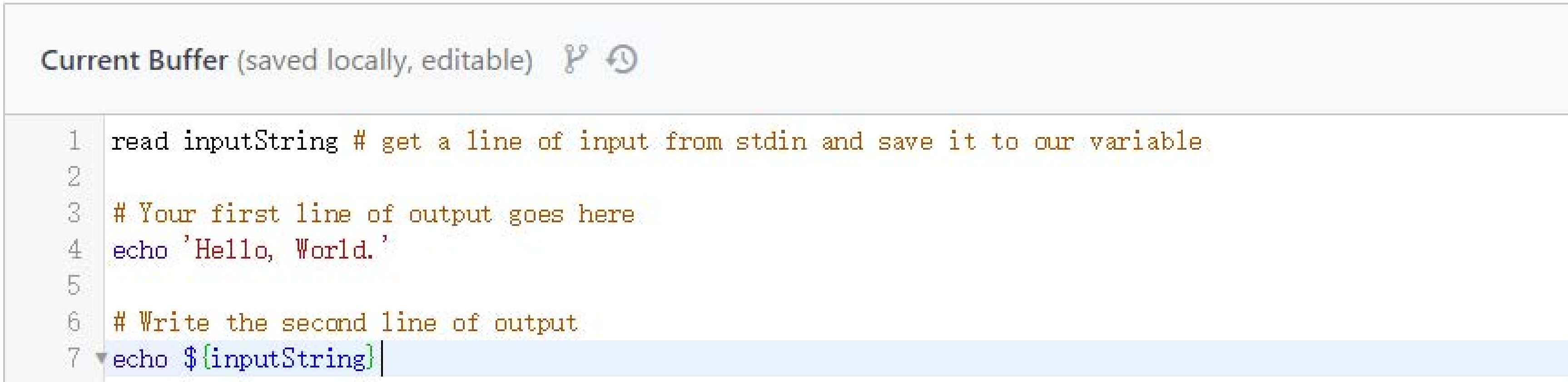
```
Welcome to 30 Days of Code!
```

Sample Output

```
Hello, World.
Welcome to 30 Days of Code!
```

Explanation

On the first line, we print the string literal `Hello, World.`. On the second line, we print the contents of the $inputString$ variable which, for this sample case, happens to be `Welcome to 30 Days of Code!`. *If you do not print the variable's contents to stdout, you will not pass the hidden test case.*

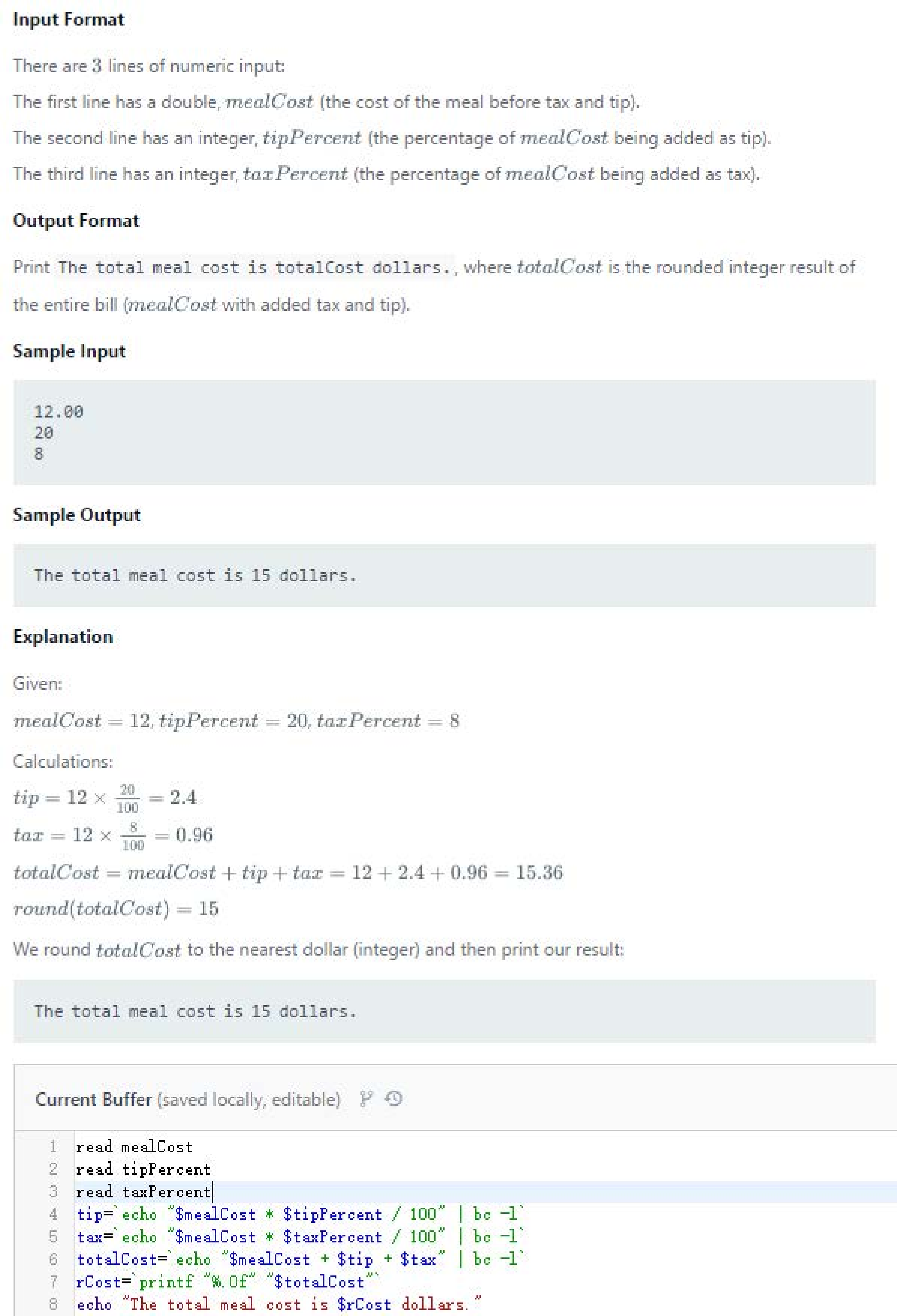


Current Buffer (saved locally, editable)

```
read inputString # get a line of input from stdin and save it to our variable

# Your first line of output goes here
echo 'Hello, World.'

# Write the second line of output
echo ${inputString}
```



Input Format

There are 3 lines of numeric input:

The first line has a double, $mealCost$ (the cost of the meal before tax and tip).

The second line has an integer, $tipPercent$ (the percentage of $mealCost$ being added as tip).

The third line has an integer, $taxPercent$ (the percentage of $mealCost$ being added as tax).

Output Format

Print `The total meal cost is totalCost dollars.`, where $totalCost$ is the rounded integer result of the entire bill ($mealCost$ with added tax and tip).

Sample Input

```
12.00
20
8
```

Sample Output

```
The total meal cost is 15 dollars.
```

Explanation

Given:

$mealCost = 12, tipPercent = 20, taxPercent = 8$

Calculations:

$tip = 12 \times \frac{20}{100} = 2.4$

$tax = 12 \times \frac{8}{100} = 0.96$

$totalCost = mealCost + tip + tax = 12 + 2.4 + 0.96 = 15.36$

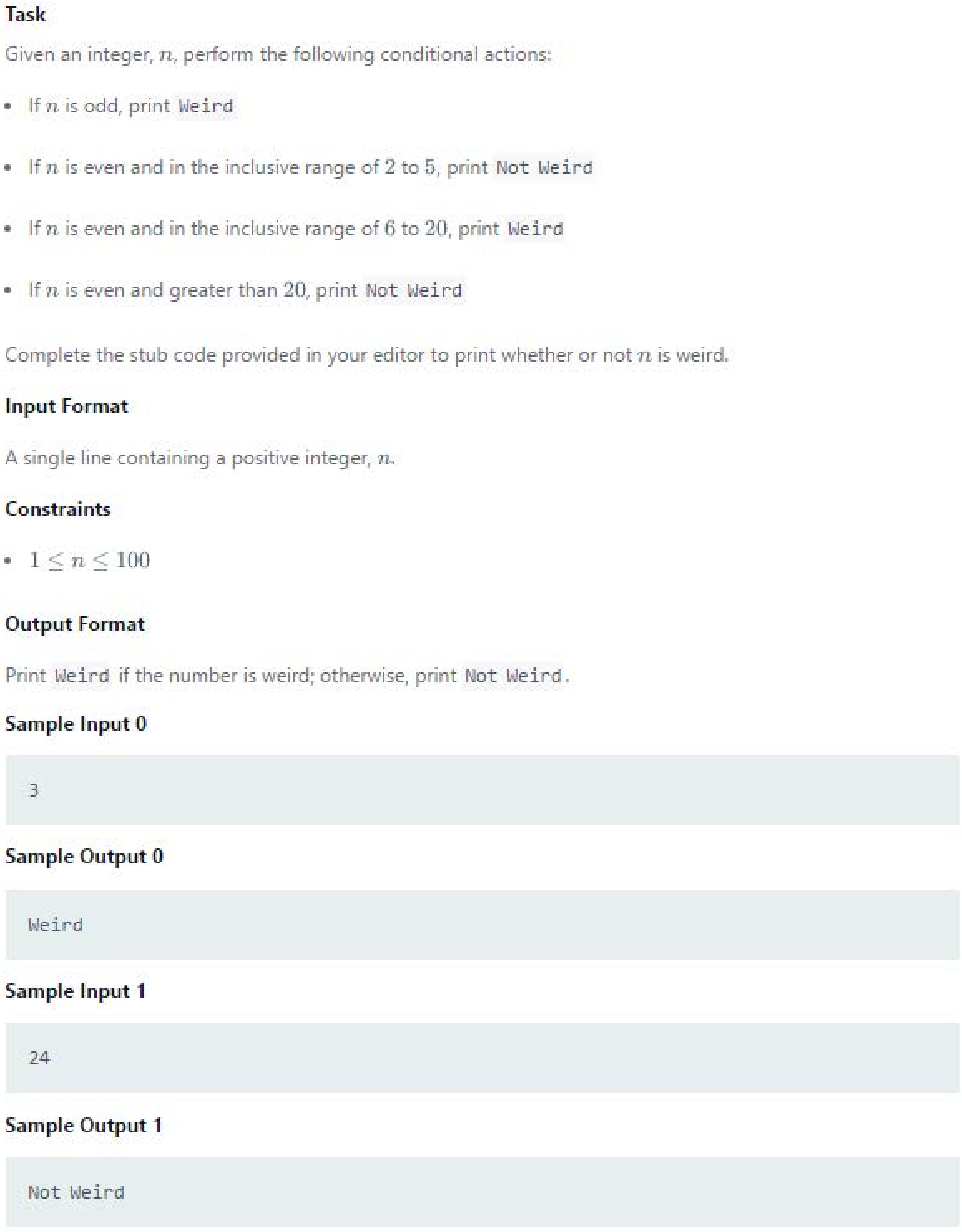
$round(totalCost) = 15$

We round $totalCost$ to the nearest dollar (integer) and then print our result:

```
The total meal cost is 15 dollars.
```

Current Buffer (saved locally, editable)

```
read mealCost
read tipPercent
read taxPercent
tip=`echo "$mealCost * $tipPercent / 100" | bc -l`
tax=`echo "$mealCost * $taxPercent / 100" | bc -l`
totalCost=`echo "$mealCost + $tip + $tax" | bc -l`
rCost=`printf "%.0f" "$totalCost"`
echo "The total meal cost is $rCost dollars."
```



Task

Given an integer, n, perform the following conditional actions:

- If n is odd, print `Weird`
- If n is even and in the inclusive range of 2 to 5, print `Not Weird`
- If n is even and in the inclusive range of 6 to 20, print `Weird`
- If n is even and greater than 20, print `Not Weird`

Complete the stub code provided in your editor to print whether or not n is weird.

Input Format

A single line containing a positive integer, n.

Constraints

- $1 \leq n \leq 100$

Output Format

Print `Weird` if the number is weird; otherwise, print `Not Weird`.

Sample Input 0

```
3
```

Sample Output 0

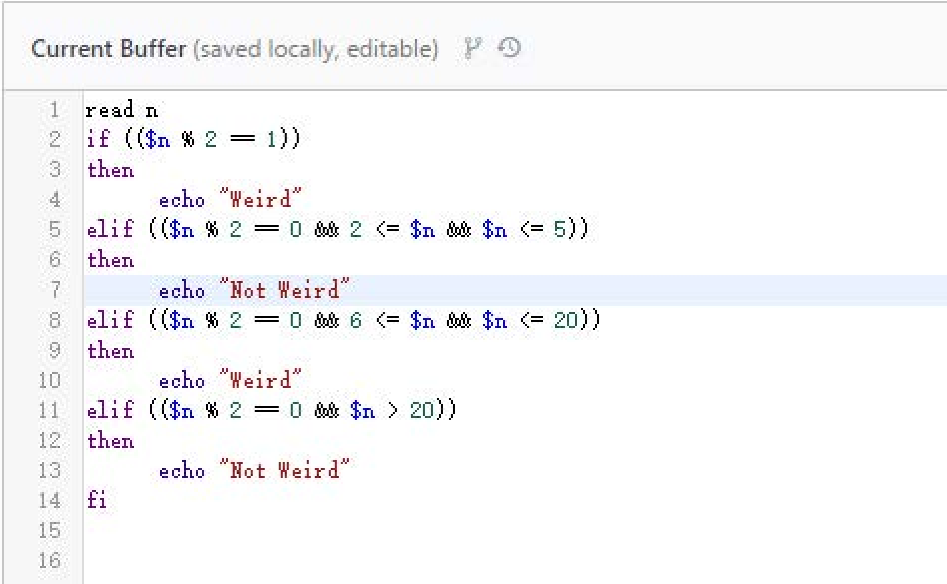
```
Weird
```

Sample Input 1

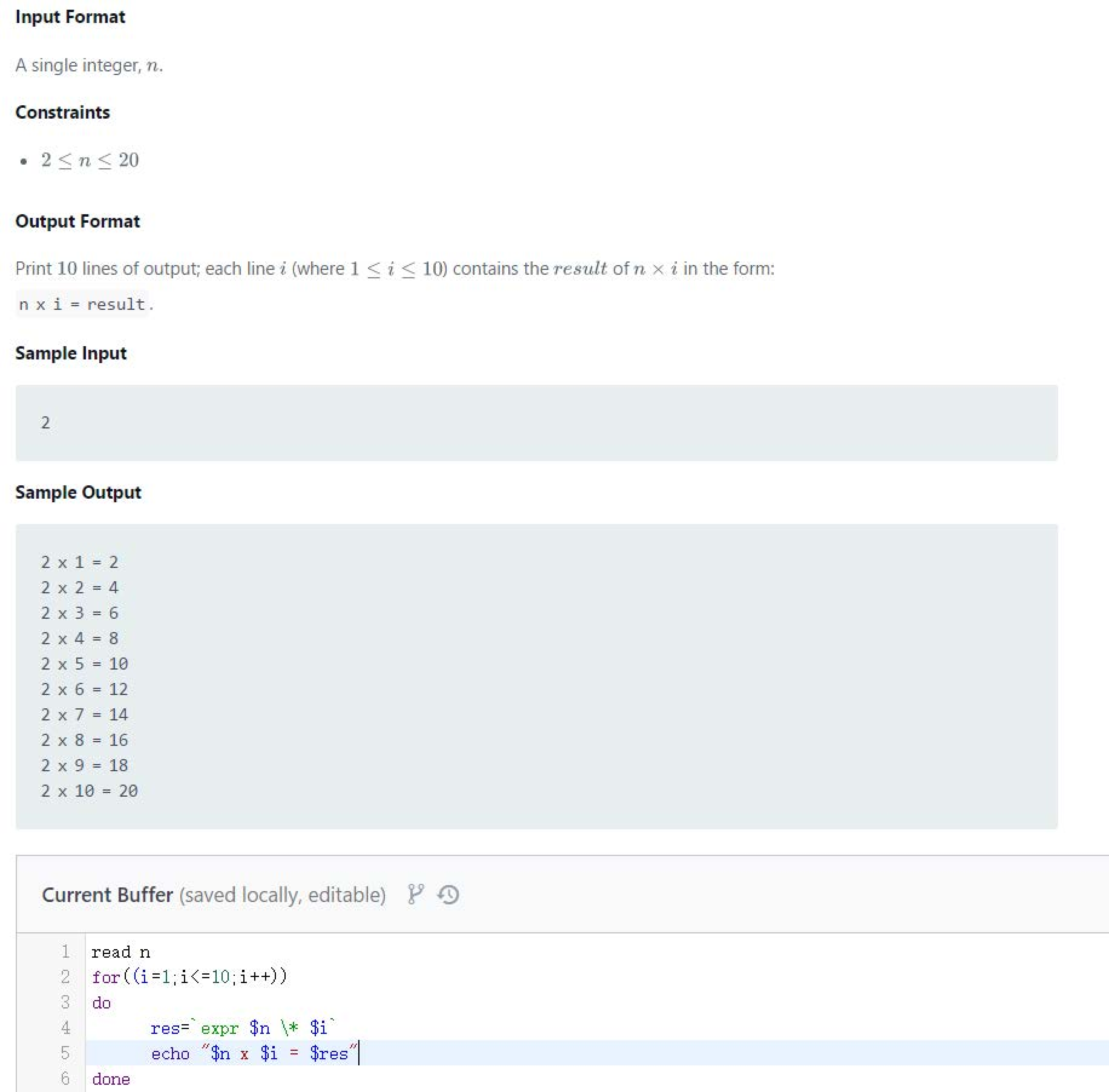
```
24
```

Sample Output 1

```
Not Weird
```



```
read n
if (($n % 2 == 1))
then
        echo "Weird"
elif (($n % 2 == 0 && 2 <= $n && $n <= 5))
then
        echo "Not Weird"
elif (($n % 2 == 0 && 6 <= $n && $n <= 20))
then
        echo "Weird"
elif (($n % 2 == 0 && $n > 20))
then
        echo "Not Weird"
fi
```



Input Format

A single integer, n.

Constraints

- $2 \leq n \leq 20$

Output Format

Print 10 lines of output; each line i (where $1 \leq i \leq 10$) contains the $result$ of $n \times i$ in the form:

`n x i = result`.

Sample Input

```
2
```

Sample Output

```
2 x 1 = 2
2 x 2 = 4
2 x 3 = 6
2 x 4 = 8
2 x 5 = 10
2 x 6 = 12
2 x 7 = 14
2 x 8 = 16
2 x 9 = 18
2 x 10 = 20
```

Current Buffer (saved locally, editable)

```
read n
for((i=1;i<=10;i++))
do
        res=`expr $n \* $i`
        echo "$n x $i = $res"
done
```