i) else if

ii) nested if else

iii) character

iv) ASCII

v) for loop

Character

Once upon a time, there was a program that took **two** integers (**a** and **b**) as input from the user. The program performed different operations based on the values of **a** and **b**.

If **a** was greater than **100** and **b** was greater than **10**, the program printed **a is above 100 and b is above 10**.If this condition was not satisfied, the program checked if **a** was greater than **50** and **b** was greater than **50**, in which case it printed **a is above 50 and b is above 50**. If this condition also didn't match, the program checked if **a** was greater than **20** and **b** was greater than **100**, and printed **a is above 20 and b is above 100** if this condition was met. If none of these conditions were met, the program simply printed **None**.

Would you like to write a program that performs these operations using your programming skills?

input a, b

x if (a > 100 && b > 10)

x if (a > 50 && b > 50)

x if (a > 20 && b > 100)

None

Take in a character as an input and manipulate it as given under

1. Condition 1: If the entered character is a **small-case** character, then

A. If the character is from character **a** and till the character **w**, both **a** and **w** included, then jump **three** times to **right** and print the resulting character as explained in the example below,

For eg, if **a** is given then print **d**, If **b** is given then print the character **e**, If c is given then print the character **f**, If **w** is given then print the character **z**.

B. Else print the string **Can't jump**.

2. Condition 2: If the entered character is a **capital-case** character, then

A. If the character is from character **D** and till the character **Z**, both **D** and **Z** included, then jump **three** times to **left** and print the resulting character as explained in the example below,

For eg, if **D** is given then print **A**, If **E** is given then print the character **B**, If F is given then print the character **C**, If **Z** is given then print the character **W**.

B. Else print the string **Can't jump**.

Input Format

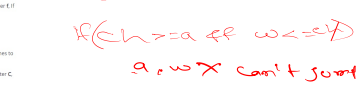
For each test case, you will get a **ch** as an character input from the user.

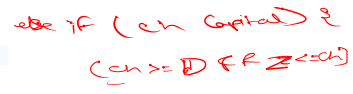
Constraints

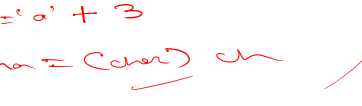
```
character input
```

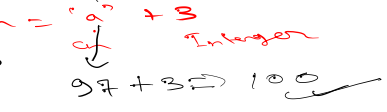
Output Format

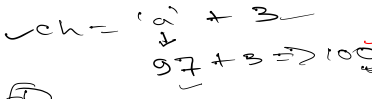
Print according to the conditions given

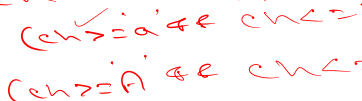


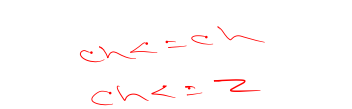


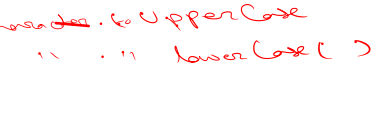












In this challenge, you must read an integer, a double, and a String from stdin, then print the values according to the instructions in the Output Format section below. To make the problem a little easier, a portion of the code is provided for you in the editor.

Note: We recommend completing Java Stdin and Stdout I before attempting this challenge.

Input Format

There are three lines of input:

1. The first line contains an integer.
2. The second line contains a double.
3. The third line contains a String.

Output Format

There are three lines of output:

1. On the first line, print `String:` followed by the unaltered String read from stdin.
2. On the second line, print `Double:` followed by the unaltered double read from stdin.
3. On the third line, print `Int:` followed by the unaltered integer read from stdin.

To make the problem easier, a portion of the code is already provided in the editor.

Note: If you use the nextLine() method immediately following the nextInt() method, recall that nextInt() reads integer tokens; because of this, the last newline character for that line of integer input is still queued in the input buffer and the next nextLine() will be reading the remainder of the integer line (which is empty).

Sample Input

```
42
3.1415
Welcome to HackerRank's Java tutorials!
```

Sample Output

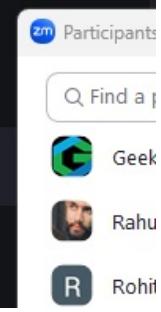
```
String: Welcome to HackerRank's Java tutorials!
Double: 3.1415
```

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int intVal = sc.nextInt();
        double doubleVal =sc.nextDouble();
        sc.nextLine();
        String stringVal =sc.nextLine();

        System.out.println("String: "+stringVal);
        System.out.println("Double: "+doubleVal);
        System.out.println("Int: "+intVal);
    }
}
```



Once upon a time, there was a young programmer named Ben who was passionate about coding. One day, his mentor challenged him to write a program that would print the first **N** multiples of **11**.

Ben approached the problem by using a loop to iterate over the numbers and multiplying each number by **11** to generate the multiples. As the loop ran, he printed each multiple to the console, ensuring that only the first **N** multiples were printed.

After a few attempts, Ben was able to complete the program successfully. He was thrilled to have solved the challenge and was proud of the new coding skills he had acquired.

Input Format

A single line take **N** as a input from user.

Constraints

```
2 <= N <= 100
```

Output Format

Print all the multiples of **11** in a single line such that each multiple of 11 should be space separated.

Sample Input 0

```
10
```

Sample Output 0

```
11 22 33 44 55 66 77 88 99 110
```

Explanation 0

Submitted Code

Language: Java 15

```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner sc = new Scanner(System.in);
        int n =sc.nextInt();

        for(int i =1;i<=n;i++){
            //int ans = 11*i;
            System.out.print(11* i+" ");
        }
    }
}
```