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
Int: 42
```


Congratulations!

You have passed the sample test cases. Click the submit button to run your code against all the test cases.

Sample Test case 0

Sample Test case 1

```
2  3.1415
3  Welcome to HackerRank's Java tutorials!
```

Your Output (stdout)

```
1  String: Welcome to HackerRank's Java tutorials!
2  Double: 3.1415
3  Int: 42
```

Expected Output Download

```
1  String: Welcome to HackerRank's Java tutorials!
2  Double: 3.1415
3  Int: 42
```

Problem | Submissions | Leaderboard | Discussions | Editorial

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
Int: 42
```

Change Theme Language Java 7

```
1  import java.util.Scanner;
2
3
4  public class Solution {
5
6      public static void main(String[] args) {
7          Scanner scan = new Scanner(System.in);
8
9          int i = scan.nextInt();
10
11         // Write your code here.
12         double d=scan.nextDouble();
13         scan.nextLine();
14         String s=scan.nextLine();
15
16         System.out.println("String: " + s);
17         System.out.println("Double: " + d);
18         System.out.println("Int: " + i);
19     }
20 }
```

Line: 8 Col: 8

Upload Code as File Test against custom input Run Code Submit Code

Problem | Submissions | Leaderboard | Discussions | Editorial

Objective

In this challenge, we're going to use loops to help us do some simple math.

Task

Given an integer, N, print its first 10 multiples. Each multiple $N \times i$ (where $1 \leq i \leq 10$) should be printed on a new line in the form: `N x i = result`.

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
```

```
24      }
25  }
26
```

Line: 19 Col: 46

Upload Code as File | Test against custom input | Run Code | Submit Code

Congratulations!

You have passed the sample test case. Click the submit button to run your code against all the test cases.

Sample Test case 0

Input (stdin) Download

```
2
```

Your Output (stdout)

```
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
```

Problem | Submissions | Leaderboard | Discussions | Editorial

Objective

In this challenge, we're going to use loops to help us do some simple math.

Task

Given an integer, N, print its first 10 multiples. Each multiple $N \times i$ (where $1 \leq i \leq 10$) should be printed on a new line in the form: `N x i = result`.

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
```

Change Theme Language Java 7

```
1  import java.io.*;
2  import java.math.*;
3  import java.security.*;
4  import java.text.*;
5  import java.util.*;
6  import java.util.concurrent.*;
7  import java.util.regex.*;
8
9
10
11 public class Solution {
12     public static void main(String[] args) throws IOException {
13         BufferedReader bufferedReader = new BufferedReader(new InputStreamReader(System.in));
14
15         int N = Integer.parseInt(bufferedReader.readLine().trim());
16
17         for(int i=1;i<=10;i++)
18         {
19             System.out.println(N+" x "+i+" = "+(N*i));
20         }
21
22         bufferedReader.close();
23
24     }
25 }
26
```

Line: 19 Col: 46

Upload Code as File | Test against custom input | Run Code | Submit Code

Problem

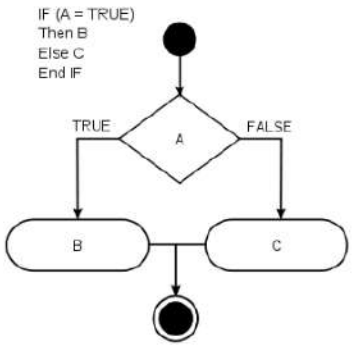
Submissions

Leaderboard

Discussions

Editorial

In this challenge, we test your knowledge of using if-else conditional statements to automate decision-making processes. An if-else statement has the following logical flow:



Source: Wikipedia

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

```
35    }
36  }
37  }
38
```

Line: 6 Col: 31

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 10.00 points!
You are now 5 points away from the 1st star for your java badge.

80% 20/25

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5
Test case 6

Compiler Message

Success

Input (stdin) Download

```
3
```

Expected Output Download

```
Weird
```

Problem

A palindrome is a word, phrase, number, or other sequence of characters which reads the same backward or forward.

Given a string A, print `Yes` if it is a palindrome, print `No` otherwise.

Constraints

- A will consist at most 50 lower case english letters.

Sample Input

```
madam
```

Sample Output

```
Yes
```

Submissions | Leaderboard | Discussions | Editorial

Line: 13 Col: 35

UploadCodeasFile | Test against custom input | Run Code | Submit Code

You have earned 10.00 points!
You are now 15 points away from the 1st star for your java badge.
40% 10/25

Congratulations
You solved this challenge. Would you like to challenge your friends? Next Challenge

Test case 0
Test case 1
Test case 2
Test case 3
Test case 4
Test case 5

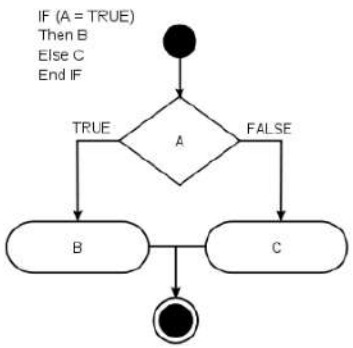
Compiler Message
Success

Input (stdin) Download
1 madam

Expected Output Download
1 Yes

Problem | Submissions | Leaderboard | Discussions | Editorial

In this challenge, we test your knowledge of using if-else conditional statements to automate decision-making processes. An if-else statement has the following logical flow:



Source: Wikipedia

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

Change Theme Language Java 7

```java
import java.io.*;
import java.math.*;
import java.security.*;
import java.text.*;
import java.util.*;
import java.util.concurrent.*;
import java.util.regex.*;

public class Solution {



    private static final Scanner scanner = new Scanner(System.in);

    public static void main(String[] args) {
        int n = scanner.nextInt();
        scanner.skip("(\r\n|[\n\r\u2028\u2029\u0085])?");

        scanner.close();
        if(n%2!=0)
        {
            System.out.println("Weird");
        }
        else if(n%2==0 && n>=5 && n<=2)
        {
            System.out.println("Not Weird");
        }
        else if(n%2==0 && n>=6 && n<=20)
        {
            System.out.println("Weird");
        }
        else
        {
            System.out.println("Not Weird");
        }
    }
}

```

Line: 15 Col: 45

Upload Code as File | Test against custom input | Run Code | Submit Code

Problem | Submissions | Leaderboard | Discussions | Editorial

Two strings, a and b, are called anagrams if they contain all the same characters in the same frequencies. For this challenge, the test is not case-sensitive. For example, the anagrams of `CAT` are `CAT`, `ACT`, `tac`, `TCA`, `aTC`, and `CtA`.

Function Description

Complete the isAnagram function in the editor.

IsAnagram has the following parameters:

- string a: the first string
- string b: the second string

Returns

- boolean: If a and b are case-insensitive anagrams, return true. Otherwise, return false.

Input Format

The first line contains a string a.
The second line contains a string b.

Constraints

- $1 \leq length(a), length(b) \leq 50$
- Strings a and b consist of English alphabetic characters.
- The comparison should NOT be case sensitive.

Sample Input 0

```
anagram
margana
```

Sample Output 0

```
Anagrams
```

Explanation 0

Change Theme | Language: Java 7

```
import java.util.Scanner; ···

    static boolean isAnagram(String a, String b) {
        // Complete the function
        a = a.toLowerCase();
        b = b.toLowerCase();

        char []a1 = a.toCharArray();
        char []b1 = b.toCharArray();

        java.util.Arrays.sort(a1);
        java.util.Arrays.sort(b1);

        if(a1.length!= b1.length) return false;
        int n = a1.length; // we can write int n = b1.length as well
        for(int i = 0;i < n; i++){
            if(a1[i] != b1[i])
                return false;
        }
        return true;
    }

    public static void main(String[] args) {···
```

Line: 8 Col: 14

Upload Code as File | Test against custom input | Run Code | Submit Code

```
class Sports{
    String getName(){
        return "Generic Sports";
    }
    void getNumberOfTeamMembers(){
        System.out.println( "Each team has n players in " + getName() );
    }
}
```

Next, we create a Soccer class that inherits from the Sports class. We can override the getName method and return a different, subclass-specific string:

```
class Soccer extends Sports{
    @Override
    String getName(){
        return "Soccer Class";
    }
}
```

Note: When overriding a method, you should precede it with the `@Override` annotation. The parameter(s) and return type of an overridden method must be exactly the same as those of the method inherited from the supertype.

Task

Complete the code in your editor by writing an overridden getNumberOfTeamMembers method that prints the same statement as the superclass' getNumberOfTeamMembers method, except that it replaces n with 11 (the number of players on a Soccer team).

Output Format

When executed, your completed code should print the following:

```
Generic Sports
Each team has n players in Generic Sports
Soccer Class
Each team has 11 players in Soccer Class
```

```
35      }
36  }
37
```

Line: 21 Col: 46

UploadCodeasFile | Test against custom input | Run Code | Submit Code

You have earned 10.00 points!
You are now 5 points away from the 2nd star for your java badge.
80% 45/50

Congratulations
You solved this challenge. Would you like to challenge your friends? Next Challenge

Test case 0

Compiler Message

Success

Expected Output Download

```
Generic Sports
Each team has n players in Generic Sports
Soccer Class
Each team has 11 players in Soccer Class
```

```
class Sports{
    String getName(){
        return "Generic Sports";
    }
    void getNumberOfTeamMembers(){
        System.out.println( "Each team has n players in " + getName() );
    }
}
```

Next, we create a Soccer class that inherits from the Sports class. We can override the getName method and return a different, subclass-specific string:

```
class Soccer extends Sports{
    @Override
    String getName(){
        return "Soccer Class";
    }
}
```

Note: When overriding a method, you should precede it with the `@Override` annotation. The parameter(s) and return type of an overridden method must be exactly the same as those of the method inherited from the supertype.

Task

Complete the code in your editor by writing an overridden getNumberOfTeamMembers method that prints the same statement as the superclass' getNumberOfTeamMembers method, except that it replaces n with 11 (the number of players on a Soccer team).

Output Format

When executed, your completed code should print the following:

```
Generic Sports
Each team has n players in Generic Sports
Soccer Class
Each team has 11 players in Soccer Class
```

Change Theme Language Java 7

```
1  import java.util.*;
2  class Sports{
3
4      String getName(){
5          return "Generic Sports";
6      }
7
8      void getNumberOfTeamMembers(){
9          System.out.println( "Each team has n players in " + getName() );
10     }
11 }
12
13 class Soccer extends Sports{
14     @Override
15     String getName(){
16         return "Soccer Class";
17     }
18
19     // Write your overridden getNumberOfTeamMembers method here
20     void getNumberOfTeamMembers(){
21         System.out.println( "Each team has 11 players in " + getName() );
22     }
23
24 }
25
26 public class Solution{
27
28     public static void main(String []args){
29         Sports c1 = new Sports();
30         Soccer c2 = new Soccer();
31         System.out.println(c1.getName());
32         c1.getNumberOfTeamMembers();
33         System.out.println(c2.getName());
34         c2.getNumberOfTeamMembers();
35     }
36 }
```

Line: 21 Col: 46

Upload Code as File Test against custom input Run Code Submit Code

of $myArray$, we can store integers at indices 0, 1, 2, and 3. Let's say we wanted the last cell to store the number 12; to do this, we write:

```
myArray[3] = 12;
```

Similarly, we can print the contents of the last cell with the following code:

```
System.out.println(myArray[3]);
```

The code above prints the value stored at index 3 of $myArray$, which is 12 (the value we previously stored there). It's important to note that while Java initializes each cell of an array of integers with a 0, not all languages do this.

Task

The code in your editor does the following:

1. Reads an integer from stdin and saves it to a variable, n, denoting some number of integers.
2. Reads n integers corresponding to $a_0, a_1, \ldots, a_{n-1}$ from stdin and saves each integer a_i to a variable, val.
3. Attempts to print each element of an array of integers named a.

Write the following code in the unlocked portion of your editor:

1. Create an array, a, capable of holding n integers.
2. Modify the code in the loop so that it saves each sequential value to its corresponding location in the array.
   For example, the first value must be stored in a_0, the second value must be stored in a_1, and so on.

Good luck!

Input Format

The first line contains a single integer, n, denoting the size of the array.
Each line i of the n subsequent lines contains a single integer denoting the value of element a_i.

Output Format

You are not responsible for printing any output to stdout. Locked code in the editor loops through array a and prints each sequential element on a new line.

Sample Input

```
5
```

Line: 21 Col: 2

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 5.00 points!
You are now 15 points away from the 2nd star for your java badge.
40% 35/50

Congratulations
You solved this challenge. Would you like to challenge your friends? Next Challenge

Test case 0
Test case 1

Compiler Message
Success

of $myArray$, we can store integers at indices 0, 1, 2, and 3. Let's say we wanted the last cell to store the number 12; to do this, we write:

```
myArray[3] = 12;
```

Similarly, we can print the contents of the last cell with the following code:

```
System.out.println(myArray[3]);
```

The code above prints the value stored at index 3 of $myArray$, which is 12 (the value we previously stored there). It's important to note that while Java initializes each cell of an array of integers with a 0, not all languages do this.

Task

The code in your editor does the following:

1. Reads an integer from stdin and saves it to a variable, n, denoting some number of integers.
2. Reads n integers corresponding to $a_0, a_1, \ldots, a_{n-1}$ from stdin and saves each integer a_i to a variable, val.
3. Attempts to print each element of an array of integers named a.

Write the following code in the unlocked portion of your editor:

1. Create an array, a, capable of holding n integers.
2. Modify the code in the loop so that it saves each sequential value to its corresponding location in the array. For example, the first value must be stored in a_0, the second value must be stored in a_1, and so on.

Good luck!

Input Format

The first line contains a single integer, n, denoting the size of the array.
Each line i of the n subsequent lines contains a single integer denoting the value of element a_i.

Output Format

You are not responsible for printing any output to stdout. Locked code in the editor loops through array a and prints each sequential element on a new line.

Sample Input

```
5
```

Change Theme Language Java 7

```
1  import java.util.*;
2
3  public class Solution {
4
5      public static void main(String[] args) {
6
7          Scanner scan = new Scanner(System.in);
8          int n = scan.nextInt();
9          int a[]=new int[n];
10         for(int i=0;i<n;i++)
11         {
12             a[i]=scan.nextInt();
13         }
14         scan.close();
15
16         // Prints each sequential element in array a
17         for (int i = 0; i < a.length; i++) {
18             System.out.println(a[i]);
19         }
20     }
21 }
```

Line: 21 Col: 2

Upload Code as File Test against custom input Run Code Submit Code

Inbox (1,089) - deeneshkumar.v | RFMiD - Google Drive | Solve Java | HackerRank | Java Anagrams | HackerRank | Java Anagrams Discussions | Java | map in java - Google Search

hackerrank.com/challenges/java-anagrams/problem?isFullScreen=true

HackerRank Prepare > Java > Strings > Java Anagrams

Exit Full Screen View

Problem | Submissions | Leaderboard | Discussions | Editorial

Two strings, a and b, are called anagrams if they contain all the same characters in the same frequencies. For this challenge, the test is not case-sensitive. For example, the anagrams of `CAT` are `CAT`, `ACT`, `tac`, `TCA`, `aTC`, and `CtA`.

Function Description

Complete the isAnagram function in the editor.

IsAnagram has the following parameters:

- string a: the first string
- string b: the second string

Returns

- boolean: If a and b are case-insensitive anagrams, return true. Otherwise, return false.

Input Format

The first line contains a string a.
The second line contains a string b.

Constraints

- $1 \leq length(a), length(b) \leq 50$
- Strings a and b consist of English alphabetic characters.
- The comparison should NOT be case sensitive.

Sample Input 0

```
anagram
margana
```

Sample Output 0

```
Anagrams
```

Explanation 0

Line: 8 Col: 14

Upload Code as File | Test against custom input | Run Code | Submit Code

You have earned 10.00 points!
You are now 20 points away from the 2nd star for your java badge.

20% 30/50

Congratulations

You solved this challenge. Would you like to challenge your friends?

Next Challenge

Test case 10
Test case 11
Test case 12
Test case 13
Test case 14
Test case 15
Test case 16

Compiler Message

Success

Input (stdin) Download

```
anagram
margana
```

Expected Output Download

```
Anagrams
```

91°F Sunny | Search | ENG IN | 13:37 13-02-2023