# Problem 2 – Magic Sum

You are given a number **D** and several lines each holding an **integer** number. Your task is to find the **biggestsum** of 3 numbers that when **divided** by **D** has **remainder0** (**sum % D = 0**). If they are 2 or more **magicSums** print the **upper mostmagicSum.** Those sums will be called **magicSum**. Example: **D = 5** numbers: **5, 10, 22 and 15.** The biggest **magicSum** is **(5 + 10 + 15), 30 % 5 = 0.**

### Input

The input data should be read from the console. At the first line, we have an integer number **D** – the divider of all sums. The next several input lines will hold **integer** numbers. At the last line the string "**End**" stays to indicate the end of the list.

The input data will always be valid and in the format described. There is no need to check it explicitly.

### Output

Print at the console the **magicSum** in the format: "(**a + b + c) % D = 0**". Note that **a, b and c** should be printed in **order of appearance.**  Beware of **spaces**: put spaces around the "**+**", "**%**" and "**=**". In case no, **magicSum** is found, print "**No**".

### Constraints

* The input number **D** will be an integer in the range [1…1000]
* All other input numbers will be integers in the range [-10000…10000].
* The **count** of the input numbers will be in the range [3..100].
* Time limit: 0.3 sec. Memory limit: 16 MB.

### Examples

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Input** | **Output** |  | **Input** | **Output** |  | **Input** | **Output** |
| 20  14  16  10  50  60  End | (10 + 50 + 60) % 20 = 0 | 6  666  333  222  111  444  555  End | (666 + 333 + 555) % 6 = 0 | 11  12  23  34  45  56  End | No |

**import** java.util.ArrayList;

**import** java.util.Scanner;

**public** **class** MagicSum {

**public** **static** **void** main(String[] args) {

Scanner input = **new** Scanner(System.in);

ArrayList<Integer> inputNumbers = **new** ArrayList<>();

**int** divider = Integer.parseInt(input.nextLine());

**while** (**true**) {

String inputLine = input.nextLine();

**if** (inputLine.equals("End")) {

**break**;

}

inputNumbers.add(Integer.parseInt(inputLine));

}

**long** maxSum = Integer.MIN\_VALUE;

String result = "";

**int** resultCounter = 0;

**for** (**int** i = 0; i < inputNumbers.size(); i++) {

**for** (**int** i1 = i+1; i1 < inputNumbers.size(); i1++) {

**for** (**int** i2 =i1 +1; i2 < inputNumbers.size(); i2++) {

**long** sum = inputNumbers.get(i) + inputNumbers.get(i1) + inputNumbers.get(i2);

**if** (sum % divider == 0 && sum > maxSum) {

maxSum = sum;

result = String.format("(%d + %d + %d) %% %d = 0",

inputNumbers.get(i), inputNumbers.get(i1), inputNumbers.get(i2), divider);

resultCounter++;

}

}

}

}

**if** (resultCounter == 0) {

System.out.println("No");

} **else** {

System.out.println(result);

}

}

}