**Algorithms with Java: Exam Preparation**

This document defines the exam preparation for ["Algorithms – Fundamentals (Java)" course @ Software University](https://softuni.bg/trainings/2991/algorithms-fundamentals-with-java-may-2020). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/Practice/Index/2481#0).

1. **Monkey Business**

*So nothing to do right? But how is that even possible. No matter what there is always something to be done and you never get bored doing the Monkey Business.*

You have found a number **N** and now you want to see in how many ways you can combine the numbers from **1 to N** in such a way that **by using addition or subtraction you will end up with zero** as a result. Find the possible **expressions** print them on the console each on a new line then followed by the total number of solutions in the following format:

* **Total Solutions: {possibleSolutions}**

## Input

* The input will come from the console on single line.

## Output

* The output is each expression on a new line followed by the total solutions **see the examples below.**

## Constraints

* **The input will only be a single positive integer.**
* The input will be in the range **[1…25]**

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 4 | +1 -2 -3 +4  -1 +2 +3 -4  Total Solutions: 2 |
| 8 | +1 +2 +3 +4 -5 -6 -7 +8  +1 +2 +3 -4 +5 -6 +7 -8  +1 +2 -3 +4 +5 +6 -7 -8  +1 +2 -3 -4 -5 -6 +7 +8  +1 -2 +3 -4 -5 +6 -7 +8  +1 -2 -3 +4 +5 -6 -7 +8  +1 -2 -3 +4 -5 +6 +7 -8  -1 +2 +3 -4 +5 -6 -7 +8  -1 +2 +3 -4 -5 +6 +7 -8  -1 +2 -3 +4 +5 -6 +7 -8  -1 -2 +3 +4 +5 +6 -7 -8  -1 -2 +3 -4 -5 -6 +7 +8  -1 -2 -3 +4 -5 +6 -7 +8  -1 -2 -3 -4 +5 +6 +7 -8  Total Solutions: 14 |

*"In the beginning there was nothing, which exploded."*

**Algorithms with Java: Exam Preparation**

This document defines the exam preparation for ["Algorithms – Fundamentals (Java)" course @ Software University](https://softuni.bg/trainings/2991/algorithms-fundamentals-with-java-may-2020). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/Practice/Index/2481#1).

# 2. Cluster Border

*Still after decades of space voyagers we have discovered new worlds and our species has spread all over the universe. We have not forgotten our old ways and we tend to stick to them as something important and something to worship something foolish like… tradition.*

There is a war going on between different Clusters and borders have been set, which means there is border control. You are part of the [Laniakea Supercluster](https://en.wikipedia.org/wiki/Laniakea_Supercluster) border control IT team.

You have been assigned task to develop a program which maximizes the work of the administration.

There is the way in which this works all the spaceships come in line one after the other.

There are **two ways of processing the ships** the first is to check a **single ship** and the other is to check one ship but let **two pass as long as the first guaranties for the second** (which means that those two ships have to agree for that) or said in a simpler way **to pass as a pair**, however a ship can only make an agreement with the ship before him or the ship after him in the line, since **reorder is not allowed.**

But here is the problem since the paperwork (yes, the public administration will still use paper even thousands of decades from now), **takes different amount of time** for each type of ship and for each pair of ships making a deal with other ships will **not always benefit in time.**

You will be **given two integer sequences the first is the time for each ship to pass as single client**, the second one will represent the time of ships **passing if they go as pair**.

You have to find the **minimum** time of ships processing and display that then you need to print the order of the ships passing **as numbers, not times**.

## Input

The input will come from the console on two lines.

* On the first line the sequence representing the time for each ship to go through.
* On the second line the sequence representing the time for two ships combination.

## Output

First print the minimum time for processing all the ships. In the format: **"Optimal Time: {optimalTime}"**

* Then print the ships with their pair.
* If ship has no pair **"Single {shipNumber}"**
* If there is pair **"Pair of {shipNumber} and {shipNumber}"**

## Constraints

* All input lines will be **valid integers you do not need to check that.**
* The range of the sequences will be **[1…1000]**

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8 5 3 9 2 1 4 4 1 17  1 3 9 4 2 4 9 3 8 | Optimal Time: 24  Pair of 1 and 2  Single 3  Pair of 4 and 5  Pair of 6 and 7  Single 8  Pair of 9 and 10 |
| 12 24 3 1 22 1 9 1  1 50 10 12 5 8 7 | Optimal Time: 17  Pair of 1 and 2  Single 3  Single 4  Pair of 5 and 6  Pair of 7 and 8 |

# *"The universe seems neither benign nor hostile,*

# *merely indifferent to the concerns of such puny creatures as we."*

**Algorithms with Java: Exam Preparation**

This document defines the exam preparation for ["Algorithms – Fundamentals (Java)" course @ Software University](https://softuni.bg/trainings/2991/algorithms-fundamentals-with-java-may-2020). Please submit your solutions (source code) of all below described problems in [Judge](https://judge.softuni.bg/Contests/Practice/Index/2481#2).

# 3. Molecules

*You are part of science team which is on an exploration mission in newly discovered planet inhabiting the goldilock zone of a distant star (insert random digits as name here).*

You have found a peace of tissue which consist of different molecules connected in order.

The biology team want from you do develop a program which **by given molecule as a source determines to which other molecules there is no way to transport energy**. On the way to any other molecule you **may have to pass through other molecules etc**.

Print on a new line separated by spaces print the **numbers of molecules you cannot transport energy to from the start molecule**. Print them in **increasing** order.

## Input

The input will come from the console:

* On the first line the number of molecules **N**
* On the second line the number of connections between the molecules **M**
* On each **M** line the data describing the connections:

**{fromMolecule} {toMolecule}**

* On the next line single integer the start molecule number

## Output

* On the single output line print the molecules in **increasing order** to which there is no connection from start molecule.

## Constraints

* All input lines will be **valid integers you do not need to check that.**
* The range of the integers will be in the range **[1…1000]**
* The molecules number will be numbers from one increasing for each molecule.

## Examples

|  |  |
| --- | --- |
| **Input** | **Output** |
| 8  9  1 2  1 3  2 5  2 4  3 4  4 5  3 6  5 6  7 8  1 | 7 8 |
| 11  11  1 5  1 4  5 7  7 8  8 2  2 3  3 4  4 1  6 2  9 10  11 9  6 | 9 10 11 |

# *"The Cosmos is all that is or ever was or ever will be."*