Problem 9 | **6 Points**

There is No 'I' in TEAM

Program Name: set.cpp   Input File: set.dat

Introduction

As manager of a high-profile project for Give Us Venture Capital Inc. (motto: "We code the stuff that makes you money"), you are constantly under pressure from your supervisor to meet programming deadlines. You notice that some of the programmers on your team work very well alone, but other programmers work extremely well with particular members on the team. This synergy allows them to code better and faster than some of the other programmers working alone. You need to pick the three programmers that will give you the maximum amount of work units done in order to meet the next looming deadline.

Input

Input to this problem will consist of a (non-empty) series of up to 100 data sets. Each data set will be formatted according to the following description, and there will be **no blank lines** separating data sets.

A single data set has 3 components:

1. *Start line* - A single line, "START X", where X is the number of programmers to choose from (3 <= X <= 9).
2. *Programmer list* - A series of X lines. Each line lists, for a single programmer, a list of work unit integer values, delimited by a single space. The first value indicates a number of work units the programmer accomplishes if programmer 1 is on his team, the second value indicates a number of work units the programmer accomplishes if programmer 2 is on his team, and so on, up to X values.
   Each of these values, Y, will be: -500 <= Y <= 500.
3. *End line* - A single line, "END"

Output

For each data set, there will be exactly one line of output. The output will be a single line in the format "A B C D", where A, B, and C (in order from least to greatest) are the numbers for the three programmers that would accomplish the greatest number of work units if they worked together on a team. D, the number of work units a given team accomplishes, is calculated by adding the total number of work units each programmer on the team accomplishes. The number of work units each programmer on the team accomplishes is calculated by totalling the number of work units he accomplishes if each of the programmers (including himself) were on the team. There will be no "ties" (two different combinations of programmers that would result in the greatest number of work units).

Example: Input File

```
START 4
500 0 0 0
0 200 200 200
0 200 200 200
0 200 200 200
END
START 5
500 -500 -500 -500 500
-100 300 100 100 500
-100 100 300 100 500
-100 100 50 300 500
0 0 0 0 0
END
```

Output to screen

```
2 3 4 1800
2 3 5 1800
```