Programming Assignment #1

Magical Elements

Due: 23:59 Wednesday, September 25

Problem Description

There are S elements in the fantasy world: earth, water, air, fire, steel, gold, etc.

The composition of any object can be broken down into a mixture of elements.

For example, a Dragon Sword consists of 10 kg of steel and 20 kg of gold, while a King Slime is composed of 30 kg of water and 40 kg of gold.

An object's energy is calculated by summing up the mass of its containing element.

For instance, a Dragon Sword has $10 + 20 = 30$ units of energy; a King Slime has $30 + 40 = 70$ units of energy.

When two objects are combined, a new object is formed. The formed object contains both object's elements:

- If an element is only obtained from one object, its resulting mass equals the mass of the source.
- However, if an element exists in both source objects, the resulting mass is the greater of the two masses of that element.

Let's say a Dragon Sword and a King Slime are combined, forming Excalibur. (a legendary sword) The Excalibur consists of 10 kg of steel, 30 kg of water and 40 kg of gold.

You are a State Alchemist. There are N objects in your laboratory, and you know the composition of each object. You must find the maximum possible energy when combining K objects among the N objects.

The process should be performed as fast as possible, since the deadline of your annual assessment report is around the corner.

I/O Format

Use standard I/O. (stdin, stdout)

Input

The first line contains one integer T ($1 \leq T \leq 10$), representing the number of test cases T. Then T test cases follow.

For the first line of each test case, there are 3 positive integers N, K, S, as described in the problem.

Then there are N lines, the i^{th} line have S numbers $A_{i,1}, A_{i,2}, \ldots, A_{i,S}$.

$A_{i,j}$ represents the mass of the j^{th} element in the i^{th} object. $0 \leq A_{i,j} < 10^6$ for all test cases.

Your code will be graded by multiple test cases: (Time limit 1000 ms)

- 20%: $2 \leq K \leq N \leq 10000, S = 2$.
- 60%: $2 \leq K \leq N \leq 10, 1 \leq S \leq 5$
- Bonus 10%: $2 \leq K \leq N \leq 10000, 1 \leq S \leq 5$

Output

For each test case T, output the maximum possible energy and a new line after it.

Examples

Input 1

```
1
3 2 2
3 4
4 3
1 5
```

Output 1

```
9
```

Input 2

```
1
2 2 3
10 20 0
0 40 30
```

Output 2

```
80
```

Program Submission

1. Please use C/C++ and write your program in a single source file.
2. Your source file must be named as "`<Student_ID>_hw1.cpp`" and please make sure that all characters of the filename are in lower case. For example, if your student id is `106062000`, the name of your program file should be `106062000_hw1.cpp`.
3. Your program will be compiled in a GNU/Linux environment with:
   `g++ -O2 -std=c++14 <Student_ID>_hw1.cpp`
4. The source file must be uploaded directly, without compressing the file.
5. 0 points will be given to Plagiarism. NEVER SHOW YOUR CODE to others and you must write your code by yourself. If the codes are similar to other people and you can't explain your code properly, you will be identified as plagiarism.

Report

1. Your report must contain the flowchart or the pseudo code of your program. You have to describe how your approach works. The analysis of time complexity should be included as well.
2. The report filename must be "`<Student_ID>_hw1.pdf`" and please make sure that all characters of the filename are in lower case.

Hint: You can use recursion to solve this problem and get the basic points. For the bonus points, improve it further through other algorithms.

Grading Policy

You must submit both your source code and report. Remember the submission rules mentioned above, or you will be punished on your grade.

- Test cases 80%
- Report 20%
- Bonus 10%

If there are more than 10% of students pass the bonus test case. Only the first 10% of students to hand in their source code get the bonus.