

Bezahlen

Templonien ist wahrlich ein seltsamer Ort. Einerseits gibt es wundervolle Tempel, riesige antike Städte und unglaublich nette Menschen so weit das Auge reicht; andererseits herrschen dort einige zutiefst eigenartige Sitten vor. Am meisten macht Lea der Brauch zu schaffen, dass die Händler in Templonien stets mit so wenigen Münzen und Scheinen wie nur möglich zu bezahlen sind. Schlimmer noch, es gibt zahllose unterschiedliche Münzen und Scheine in der templonischen Währung und viele Händler akzeptieren auch nur manche davon. Als ausländische Touristin hat Lea damit naturgemäß sehr zu kämpfen, jedoch ist sie auch eine sehr tolerante Person und möchte die örtlichen Händler natürlich nicht verärgern. Wann immer es besonders anspruchsvoll ist, die richtigen Münzen und Scheine für einen bestimmten Betrag herauszusuchen, zückt sie daher das Telefon und ruft dich an. Kannst du sie bitte unterstützen, damit sie nicht den Zorn der Händler auf sich zieht?

Eingabe

Die erste Zeile der Eingabe enthält eine Ganzzahl t . Darauf folgen, jeweils durch eine Leerzeile getrennt, t Testfälle.

In der ersten Zeile jedes Testfall befinden sich zwei Ganzzahlen n und c . Die Münzen und Scheine in Templonien nehmen n verschiedene Werte an und Lea möchte den Betrag c bezahlen. Die nächste Zeile enthält n verschiedene Ganzzahlen v_1, \dots, v_n , die Werte der Münzen und Scheine in aufsteigender Reihenfolge. Du kannst davon ausgehen, dass es stets eine Münze mit Wert 1 gibt, und, dass Lea von jeder Sorte ausreichend viele Münzen oder Scheine zur Verfügung stehen.

Ausgabe

Gibt für jeden Testfall eine Zeile der Form “Case # i : x ” aus, wobei i bei 1 beginnend die Nummer des Testfall und x eine durch Leerzeichen getrennte Liste von n nicht-negativen Ganzzahlen a_1, \dots, a_n mit folgenden Eigenschaften ist: der Betrag c ergibt sich daraus, dass man a_1 Münzen mit Wert v_1 , a_2 Münzen oder Scheine mit Wert v_2 usw. miteinander kombiniert. Und die Anzahl an verwendeten Münzen oder Scheinen, d. h. die Summe $a_1 + a_2 + \dots + a_n$ soll unter allen möglichen Lösungen minimal sein. Wenn verschiedene Lösungen diese Eigenschaften erfüllen, kannst du eine beliebige davon ausgeben.

Beschränkungen

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq c \leq 10^5$
- $1 \leq v_i \leq 10^4$ für alle $1 \leq i \leq n$

Sample Input 1

```
2
6 29
1 2 4 5 6 8

4 43
1 6 7 13
```

Sample Output 1

```
Case #1: 0 0 0 1 0 3
Case #2: 0 5 0 1
```

Sample Input 2

```
5
4 780
1 6 35 97

9 827
1 15 17 26 64 70 79 88 98

5 598
1 21 45 64 68

2 756
1 55

5 507
1 23 33 35 58
```

Sample Output 2

```
Case #1: 4 0 0 8
Case #2: 1 0 0 0 0 2 0 0 7
Case #3: 0 1 1 3 5
Case #4: 41 13
Case #5: 0 0 2 1 7
```

Making Change

Templonia is a very strange place. On one hand, there are beautiful temples, great ancient cities and many nice people. On the other hand, the Templonians have acquired some very peculiar customs. The one which causes the biggest problem for Lea is that Templonian merchants always want their money in as few coins or notes as possible. Even worse, there are many different coins and notes for the Templonian Column, and not every merchant takes all values. Surely, the Templonian people are accustomed to this behaviour and are very good in calculating these things. But as a foreign tourist, Lea has some difficulties to say the least. A tolerant person as she is, she does not want to anger the local merchants and tries to adopt Templonian customs. Whenever she has an unusually hard problem at finding the right coins, she takes out her phone and calls you. Can you please help her not get beaten up by angry merchants?

Input

The first line of the input contains an integer t . t test cases follow, each of them separated by a blank line.

Each test case starts with a single line containing two integers n and c . n is the number of coin and note values and c is the amount of money that must be spent. The next line consists of n distinct integers v_1, \dots, v_n describing the coin/note values in increasing order. You may assume that a coin of value 1 is always available, and that Lea has as many of all the coins/notes as she needs.

Output

For each test case, output one line containing “Case # i : x ” where i is its number, starting at 1, and x is a space-separated sequence of n integers a_1, \dots, a_n , where a_i means that the optimal solution uses exactly a_i coins/notes of value v_i .

Constraints

- $1 \leq t \leq 20$
- $1 \leq n \leq 100$
- $1 \leq c \leq 10^5$
- $1 \leq v_i \leq 10^4$ for all $1 \leq i \leq n$

Sample Input 1

```
2
6 29
1 2 4 5 6 8

4 43
1 6 7 13
```

Sample Output 1

```
Case #1: 0 0 0 1 0 3
Case #2: 0 5 0 1
```

Sample Input 2

```
5
4 780
1 6 35 97

9 827
1 15 17 26 64 70 79 88 98

5 598
1 21 45 64 68

2 756
1 55

5 507
1 23 33 35 58
```

Sample Output 2

```
Case #1: 4 0 0 8
Case #2: 1 0 0 0 0 2 0 0 7
Case #3: 0 1 1 3 5
Case #4: 41 13
Case #5: 0 0 2 1 7
```