Seilbahn

Lea ist ein großer Wintersport-Fan, sie verfolgt stets die Olympischen Winterspiele und fährt auch selber gerne Ski. Eines Abends während ihres diesjährigen Skiurlaubs traf sie auf einen Architekten, der Spezialist ist für Seilbahnen, die die Touristen auf den Berg hoch befördern. Nachdem sie sich ein bisschen unterhielten, beschrieb er ihr seine neueste Herausforderung:

Das Skigebiet möchte eine neue Seilbahn entlang eines Gletschers bauen. Durch höchst komplexe Berechnungen wurde bereits ermittelt, wie viele Stützpfeiler für diese Seilbahn genau benötigt werden. Da sich der Gletscher ganz langsam bewegt (vor allem aufgrund des Klimawandels), müssen die Stützpfeiler so weit voneinander entfernt wie nur möglich gebaut werden. Zusätzlich soll die Seilbahn einen Eisspalt inmitten der Route überwinden. Der Architekt arbeitet nun daran herauszufinden, wie die Stützpfeiler am besten zu platzieren sind.

Lea, die stets Ausschau nach interessanten Aufgaben hält, erzählt dir sofort davon. Kannst du dem Architekten weiterhelfen?

Eingabe

Die erste Zeile der Eingabe enthält eine Ganzzahl t, die Anzahl der Testfälle.

Jeder Testfall besteht aus vier Ganzzahlen d, p, u und v. Dabei ist d die Gesamtlänge der geplanten Seilbahn-Route (die sich von Position 0 bis Position d erstreckt), p ist die Anzahl der benötigten Stützpfeiler, u ist die Startposition und v die Endposition des Eisspalts. Mit Ausnahme des Abschnittes zwischen u und v dürfen Stützpfeiler überall zwischen den Positionen 0 und d errichtet werden (einschließlich der Positionen 0, d, u und v). Bitte beachte, dass Stützpfeiler auch an Positionen errichtet werden können, die nicht ganzzahlig sind.

Ausgabe

Gibt für jeden Testfall eine Zeile "Case #i: x" aus, wobei i bei 1 beginnen die Nummer des Testfalls ist und x der maximale Mindestabstand zwischen den Stützpfeilern, der erreicht werden kann (also der maximal mögliche Wert x, sodass der Architekt alle Stützpfeiler platzieren kann und je zwei Stützpfeiler immer mindestens Abstand x voneinander haben). Dieser Wert soll eine absolute Genauigkeit von mindestens 10^{-4} haben.

Beschränkungen

- $1 \leq t \leq 20$
- $1 \leq d \leq 10^6$
- $2 \leq p \leq 2 \cdot 10^6$
- $0 \leq u \leq v \leq d$

Sample Input 1

```
4
2 3 1 2
3 3 0 1
9 10 5 6
9 10 5 7
```

Sample Output 1

```
Case #1: 1.0000000009
Case #2: 1.5000000007
Case #3: 1.0000000001
Case #4: 0.8333333338
```

Sample Input 2

```
20
8 2 3 7
9 5 6 6
6 2 1 6
5 5 3 3
5 3 4 5
9 2 3 9
5 5 3 4
7 3 6 6
6 5 2 4
8 3 7 7
6 3 5 6
7 4 2 5
10 3 2 2
7 2 4 4
7 4 3 6
6 3 0 5
8 5 4 5
7 2 6 7
5 3 5 5
8 5 1 3
```

Sample Output 2

```
Case #1: 8.0000000000
Case #2: 2.2500000005
Case #3: 6.0000000000
Case #4: 1.2500000006
Case #5: 2.5000000006
Case #6: 9.0000000000
Case #7: 1.0000000003
Case #8: 3.5000000008
Case #9: 1.0000000005
Case #10: 4.0000000009
Case #11: 3.0000000007
Case #12: 2.0000000006
Case #13: 5.0000000006
Case #14: 7.0000000000
Case #15: 1.5000000002
Case #16: 1.0000000005
Case #17: 2.0000000009
Case #18: 7.0000000000
Case #19: 2.5000000006
Case #20: 1.6666666670
```

Cable Car

Lea is a great fan of wintersports. She always follows the winter olympics on TV and just loves to go skiing herself. This year, she booked a room in an expensive hotel in a very exclusive ski resort called “Slippery Slopes and Hills”. One evening at the Après-Ski-Party, she met an interesting man - the architect who planned all the cable cars taking the tourists up the mountain. They talked for a bit and he described his latest problem to her.

The ski resort is trying to build a new cable car up a glacier. Through complicated computation, they even found out exactly how many posts are needed to support the cable car. Since glaciers move (albeit really slowly), the individual posts supporting the cable car have to be spaced as far apart from each other as possible. Additionally, the cable car should also span a canyon in the middle of the route. Now the architect is hard at work, trying to figure how to place the posts. Lea, who is always on the lookout for interesting problems, tells you about it. Can you help the architect?

Input

The first line of the input contains an integer t. t test cases follow.

Each test case consists of a single line of four integers d, p, u, and v, where d is the length of the route (going from 0 to d) of the cable car, p is the amount of posts that should be placed, u is the beginning point of the canyon and v the end point. Posts may be placed anywhere between 0 and d, i.e. exactly on 0, d, u, and v, but not in between u and v.

Output

For each test case, output one line containing “Case #i: x” where i is its number, starting at 1, and x is the maximal minimum distance between two posts that can be achieved with an absolute error of up to 10^{-4}. This means the maximum x such that the architect can place all the posts and no two posts are less than x apart. Each line of the output should end with a line break.

Constraints

- $1 \leq t \leq 20$
- $1 \leq d \leq 10^6$
- $2 \leq p \leq 2 \cdot 10^6$
- $0 \leq u \leq v \leq d$

Sample Input 1	Sample Output 1
4 2 3 1 2 3 3 0 1 9 10 5 6 9 10 5 7	Case #1: 1.0000000009 Case #2: 1.5000000007 Case #3: 1.0000000001 Case #4: 0.8333333338

Sample Input 2

```
20
8 2 3 7
9 5 6 6
6 2 1 6
5 5 3 3
5 3 4 5
9 2 3 9
5 5 3 4
7 3 6 6
6 5 2 4
8 3 7 7
6 3 5 6
7 4 2 5
10 3 2 2
7 2 4 4
7 4 3 6
6 3 0 5
8 5 4 5
7 2 6 7
5 3 5 5
8 5 1 3
```

Sample Output 2

```
Case #1: 8.0000000000
Case #2: 2.2500000005
Case #3: 6.0000000000
Case #4: 1.2500000006
Case #5: 2.5000000006
Case #6: 9.0000000000
Case #7: 1.0000000003
Case #8: 3.5000000008
Case #9: 1.0000000005
Case #10: 4.0000000009
Case #11: 3.0000000007
Case #12: 2.0000000006
Case #13: 5.0000000006
Case #14: 7.0000000000
Case #15: 1.5000000002
Case #16: 1.0000000005
Case #17: 2.0000000009
Case #18: 7.0000000000
Case #19: 2.5000000006
Case #20: 1.6666666670
```