

A. New Year and Hurry

1 second, 256 megabytes

Limak is going to participate in a contest on the last day of the 2016. The contest will start at 20:00 and will last four hours, exactly until midnight. There will be n problems, sorted by difficulty, i.e. problem 1 is the easiest and problem n is the hardest. Limak knows it will take him $5 \cdot i$ minutes to solve the i -th problem.

Limak's friends organize a New Year's Eve party and Limak wants to be there at midnight or earlier. He needs k minutes to get there from his house, where he will participate in the contest first.

How many problems can Limak solve if he wants to make it to the party?

Input

The only line of the input contains two integers n and k ($1 \leq n \leq 10$, $1 \leq k \leq 240$) — the number of the problems in the contest and the number of minutes Limak needs to get to the party from his house.

Output

Print one integer, denoting the maximum possible number of problems Limak can solve so that he could get to the party at midnight or earlier.

input
3 222
output
2

input
4 190
output
4

input

7 1

output

7

In the first sample, there are 3 problems and Limak needs 222 minutes to get to the party. The three problems require 5, 10 and 15 minutes respectively. Limak can spend $5 + 10 = 15$ minutes to solve first two problems. Then, at 20:15 he can leave his house to get to the party at 23:57 (after 222 minutes). In this scenario Limak would solve 2 problems. He doesn't have enough time to solve 3 problems so the answer is 2.

In the second sample, Limak can solve all 4 problems in $5 + 10 + 15 + 20 = 50$ minutes. At 20:50 he will leave the house and go to the party. He will get there exactly at midnight.

In the third sample, Limak needs only 1 minute to get to the party. He has enough time to solve all 7 problems.

B. Onde está o Mármore?

1 segundo, 256 megabytes

Raju e Meena adoram jogar um jogo diferente com pequenas peças de mármore. Eles têm um monte destas peças com números escritos neles. No início do jogo, Raju coloca estes pequenos mármorees um após outro em ordem ascendente de números escritos neles. Então Meena pede a Raju para encontrar o primeiro mármore com um certo número. Ele deve contar 1...2...3. Raju ganha um ponto por cada resposta correta e Meena ganha um ponto se Raju falha. Depois de um número fixo de tentativas, o jogo termina e o jogador com o máximo de pontos vence. Hoje é sua chance de jogar com Raju. Sendo um/a cara esperto/a, você tem a seu favor o computador. Mas não subestime Meena, ela escreveu um programa para monitorar quanto tempo você levará para dar todas as respostas. Portanto, agora escreva o programa, que ajudará você em seu desafio com Raju.

Input

A entrada contém vários casos de teste, mas o total de casos é menor ou igual a 30. Cada caso de teste inicia com dois inteiros: N que é o número de mármorees e Q que é o número de consultas que Meena deseja fazer. As próximas N linhas conterão os números escritos em cada um dos N mármorees. Os números destes mármorees não tem qualquer ordem em particular. As seguintes Q linhas irão conter Q consultas. Nenhum dos números da entrada é maior do que 10000 e nenhum deles é negativo. A entrada é terminada por um caso de teste onde $N = 0$ e $Q = 0$.

Output

Para cada caso de teste de saída deve haver um número serial do caso de teste. Para cada consulta, escreva uma linha de saída. O formato desta linha dependerá se o número consultado estiver ou não escrito em um dos mármorees. Os dois diferentes formatos são descritos abaixo:

- "x found at y", se o primeiro mármore x foi encontrado na posição y . Posições são numeradas de 1, 2, ... a N .
- "x not found", se o mármore com o número x não estiver presente.

input
4 1 2 3 5 1 5 5 2 1 3 3 3 1 2 3 0 0
output
CASE# 1: 5 found at 4 CASE# 2: 2 not found 3 found at 3

C. Hamburgers

1 second, 256 megabytes

Polycarpus loves hamburgers very much. He especially adores the hamburgers he makes with his own hands. Polycarpus thinks that there are only three decent ingredients to make hamburgers from: a bread, sausage and cheese. He writes down the recipe of his favorite "Le Hamburger de Polycarpus" as a string of letters 'B' (bread), 'S' (sausage) and 'C' (cheese). The ingredients in the recipe go from bottom to top, for example, recipe "BSCBS" represents the hamburger where the ingredients go from bottom to top as bread, sausage, cheese, bread and sausage again.

Polycarpus has r rubles and he is ready to shop on them. What maximum number of hamburgers can he cook? You can assume that Polycarpus cannot break or slice any of the pieces of bread, sausage or cheese. Besides, the shop has an unlimited number of pieces of each ingredient.

The first line of the input contains a non-empty string that describes the recipe of "Le Hamburger de Polycarpus". The length of the string doesn't exceed 100, the string contains only letters 'B' (uppercase English B), 'S' (uppercase English S) and 'C' (uppercase English C).

Please, do not write the %lld specifier to read or write 64-bit integers in C++. It is preferred to use the cin, cout streams or the %I64d specifier.

Print the maximum number of hamburgers Polycarpus can make. If he can't make any hamburger, print 0.

input
BBBSSC 6 4 1 1 2 3 4
output
2

input
BBC 1 10 1 1 10 1 21
output
7

input
BSC 1 1 1 1 1 3 1000000000000
output
2000000000001

1 segundo, 256 megabytes

[illegible]

Input

A primeira linha da entrada contém dois inteiros N ($1 \leq N \leq 10^5$) e F ($1 \leq F \leq 10^8$), indicando o número de cápsulas e o número de moedas que Fan Chi'ih quer produzir, respectivamente. A segunda linha contém N inteiros C_i ($1 \leq C_i \leq 10^6$), para $1 \leq i \leq N$, representando os ciclos de cada cápsula. Em todos os casos de teste, a resposta é sempre menor ou igual a 10^8 dias. Em todos os casos de teste, o número de moedas produzido, no total, após 10^8 dias, é sempre menor ou igual a 10^9 .

Output

Imprima um inteiro, representando o número mínimo de dias para que as cápsulas produzam, no total, pelo menos F moedas.

input
3 12 3 7 2
output
14

input
10 100 17 13 20 10 12 16 10 13 13 10
output
130

E. Building Construction

1 second, 256 megabytes

The construction of the new university complex is almost over. However, it seems like a minor detail has been overlooked: The height of the buildings doesn't quite match. Because each building has been made of a different material, the cost of adding or removing new layers of bricks can be different for each building. In order to fix this mistake in such a way that it costs less, the engineering department has hired you.

You are given n buildings of heights $h_1, h_2, \dots, h_n$. For each one, there's a cost of c_i for either removing or adding a new layer of bricks to the building. Your job is to find a way to make sure that all the N buildings have the same height with the minimum cost.

Input

The first line of input contains an integer t ($1 \leq t \leq 15$) — the number of test cases.

The first line of each test case contains an integer n ($1 \leq n \leq 10000$) — the number of buildings.

The second line contains n integers $h_1, h_2, \dots, h_n$ ($0 \leq h_i \leq 10000$) — the heights of the buildings.

The third line contains n integers $c_1, c_2, \dots, c_n$ ($0 \leq c_i \leq 10000$) — the cost of adding or removing a layer of bricks to the corresponding building.

Output

For each test case, output the minimum cost to make all the buildings have the same height.

input
1 3 1 2 3 10 100 1000
output
120

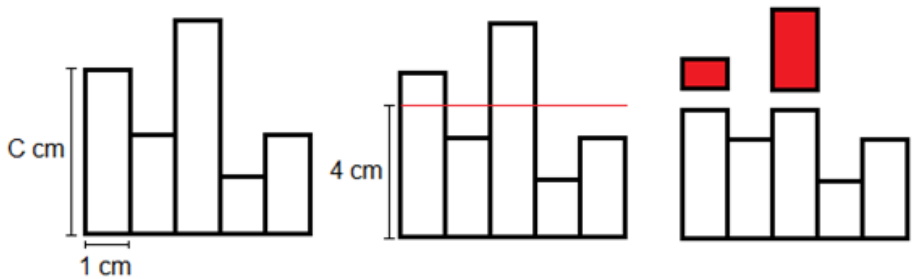
F. Ajude Seu Madruga

1 segundo, 256 megabytes

Seu Madruga finalmente arrumou um emprego, é sua grande chance de pagar os 14 meses de aluguel que estão atrasados. Seu cargo é o de cortador de papel e o salário é excelente.

Madruga irá receber N tiras retangulares de papel com 1 cm de largura e C cm de comprimento. As tiras devem ser colocadas uma ao lado da outra de forma que suas bases estejam alinhadas. A tarefa de seu Madruga será, com apenas um corte em linha reta, paralelo à base, fazer com que a soma das áreas das tiras cortadas seja igual a $A \text{ cm}^2$.

Veja a ilustração abaixo com $N = 5$ e as tiras com 5, 3, 6, 2 e 3 cm de comprimento, respectivamente, para um $A = 3 \text{ cm}^2$.



Com um corte feito a uma altura de 4 cm em relação a base, a área resultante, pintada de vermelho, é exatamente igual a $A \text{ cm}^2$. Sua tarefa é descobrir essa altura H e ajudar Seu Madruga a se manter no emprego.

Input

Haverá diversos casos de testes. A primeira linha de cada caso inicia com dois inteiros N ($1 \leq N \leq 10^5$) e A ($1 \leq A \leq 10^9$) representando, respectivamente, a quantidade de tiras e a área resultante esperada. A próxima linha conterá N inteiros, representando o comprimento C_i ($1 \leq C_i \leq 10^4$) de cada tira.

A entrada termina com $N = A = 0$, a qual não deve ser processada.

Output

Para cada caso, exiba uma única linha, a altura H do corte que Seu Madruga deve fazer para que a soma da área das tiras cortadas seja igual a $A \text{ cm}^2$. Use 4 casas decimais após o ponto. Será tolerada uma diferença de 10^{-5} .

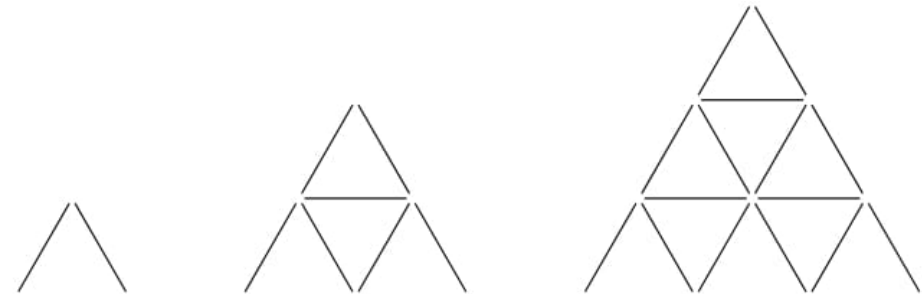
Exiba :D, caso nenhum corte seja necessário, ou -.- caso seja impossível.

input
5 3 5 3 6 2 3 4 14 2 5 2 5 0 0
output
4.0000 :D

G. Card Constructions

1 second, 256 megabytes

A card pyramid of height 1 is constructed by resting two cards against each other. For $h > 1$, a card pyramid of height h is constructed by placing a card pyramid of height $h - 1$ onto a base. A base consists of h pyramids of height 1, and $h - 1$ cards on top. For example, card pyramids of heights 1, 2, and 3 look as follows:



You start with n cards and build the tallest pyramid that you can. If there are some cards remaining, you build the tallest pyramid possible with the remaining cards. You repeat this process until it is impossible to build another pyramid. In the end, how many pyramids will you have constructed?

Input

Each test consists of multiple test cases. The first line contains a single integer t ($1 \leq t \leq 1000$) — the number of test cases. Next t lines contain descriptions of test cases.

Each test case contains a single integer n ($1 \leq n \leq 10^9$) — the number of cards.

It is guaranteed that the sum of n over all test cases does not exceed 10^9 .

Output

For each test case output a single integer — the number of pyramids you will have constructed in the end.

input
5 3 14 15 24 1
output
1 2 1 3 0

In the first test, you construct a pyramid of height 1 with 2 cards. There is 1 card remaining, which is not enough to build a pyramid.

In the second test, you build two pyramids, each of height 2, with no cards remaining.

In the third test, you build one pyramid of height 3, with no cards remaining.

In the fourth test, you build one pyramid of height 3 with 9 cards remaining. Then you build a pyramid of height 2 with 2 cards remaining. Then you build a final pyramid of height 1 with no cards remaining.

In the fifth test, one card is not enough to build any pyramids.

H. Sagheer and Nubian Market

2 seconds, 256 megabytes

On his trip to Luxor and Aswan, Sagheer went to a Nubian market to buy some souvenirs for his friends and relatives. The market has some strange rules. It contains n different items numbered from 1 to n . The i -th item has base cost a_i Egyptian pounds. If Sagheer buys k items with indices $x_1, x_2, \dots, x_k$, then the cost of item x_j is $a_{x_j} + x_j \cdot k$ for $1 \leq j \leq k$. In other words, the cost of an item is equal to its base cost in addition to its index multiplied by the factor k .

Sagheer wants to buy as many souvenirs as possible without paying more than S Egyptian pounds. Note that he cannot buy a souvenir more than once. If there are many ways to maximize the number of souvenirs, he will choose the way that will minimize the total cost. Can you help him with this task?

Input

The first line contains two integers n and S ($1 \leq n \leq 10^5$ and $1 \leq S \leq 10^9$) — the number of souvenirs in the market and Sagheer's budget.

The second line contains n space-separated integers $a_1, a_2, \dots, a_n$ ($1 \leq a_i \leq 10^5$) — the base costs of the souvenirs.

Output

On a single line, print two integers k, T — the maximum number of souvenirs Sagheer can buy and the minimum total cost to buy these k souvenirs.

input
3 11 2 3 5
output
2 11

input
4 100 1 2 5 6

output
4 54

input
1 7
7
output
0 0

In the first example, he cannot take the three items because they will cost him [5, 9, 14] with total cost 28. If he decides to take only two items, then the costs will be [4, 7, 11]. So he can afford the first and second items.

In the second example, he can buy all items as they will cost him [5, 10, 17, 22].

In the third example, there is only one souvenir in the market which will cost him 8 pounds, so he cannot buy it.

I. Cabo de Guerra

1 segundo, 256 megabytes

Na Final Nacional da Maratona de Programação de 2013, em Uberlândia, durante um período de lazer, o Prof. Carlinhos (USP) propôs uma atividade a todos os estudantes. Ele primeiro organizou os estudantes em ordem lexicográfica crescente, considerando apenas o primeiro nome e desconsiderando diacríticos. Em seguida, sorteou um estudante e montou duas equipes, A e B: a equipe A seria formada por todos os estudantes na ordem até o estudante sorteado, inclusive; a equipe B seria formada por todos os estudantes na ordem a partir do estudante sorteado, exclusive. As duas equipes, então, competiriam num cabo de guerra tradicional, e os vencedores ganhariam um café.

Muitas coisas curiosas o Prof. Carlinhos percebeu naquele dia:

- A força de cada estudante, quando posicionado imediatamente em frente ao adversário (posição 1), era exatamente igual à soma dos

- valores correspondentes aos caracteres de seu primeiro nome na tabela ASCII. Dessarte, a força do estudante Leandro, se ficasse na posição 1 de uma equipe (fosse A ou fosse B), serial igual a:
- $$709 = 76 + 101 + 97 + 110 + 100 + 114 + 111 = \text{'L'} + \text{'e'} + \text{'a'} + \text{'n'} + \text{'d'} + \text{'r'} + \text{'o'}$$
- Quanto mais distante da equipe adversária, menos intimidado — e, portanto, mais forte — ficava um estudante. Mais especificamente, um estudante na posição 2 de uma equipe tinha o dobro da força que teria se ficasse na posição 1. Na posição 3, o triplo, e assim por diante. As posições de ambas as equipes são numeradas a partir da posição mais próxima da equipe adversária, começando a contagem em 1. Por exemplo, se o estudante Leandro ficasse na posição 3 de uma equipe, teria força igual a $3 \cdot 709 = 2127$.
 - A força de uma equipe era igual à soma das forças de cada um de seus integrantes. Não obstante, se a força da equipe A fosse maior que a da equipe B, a equipe A certamente ganharia. Por outro lado, se a força da equipe A fosse menor que a da equipe B, a equipe B certamente ganharia. Finalmente, se as forças de ambas as equipes fossem iguais, haveria um empate.

Existe algum estudante que o Prof. Carlinhos poderia sortear que fizesse as equipes A e B empatarem?

Input

A entrada consiste de vários casos de teste. A primeira linha de cada caso de teste é composta por um único inteiro N ($1 \leq N \leq 10^5$), o qual representa o número de estudantes. Seguem, então, N linhas, cada uma contendo o primeiro nome de um estudante. Os nomes dos estudantes são fornecidos segundo a ordem lexicográfica crescente, e no mínimo 1 e no máximo 10 letras do alfabeto latino compõem o nome de um estudante. Não há num mesmo caso de teste dois estudantes com o mesmo primeiro nome, e a primeira letra de um nome é sempre maiúscula, sendo as demais minúsculas. $N = 0$ finaliza a entrada.

Output

Imprima o nome do estudante que, se sorteado, faria as equipes A e B empatarem. Se não há tal estudante, imprima a linha:
"Impossibilidade de empate." (sem as aspas).

input

```
9
Ana
Bruna
Cro
Digory
Emerson
Fiaror
Geomar
Iago
Zacarias
14
Aule
Este
Lorien
Mandos
Manwe
Nessa
Nienna
Orome
Tulkas
Ulmo
Vaire
Vana
Varda
Yavanna
0
```

output

```
Emerson
Impossibilidade de empate.
```