01

Some of us called Anik a master of Java. Anik also agree on this . But once a stranger challenged Anik, if anik could solve a problem he(stngr.) also will be agreed with them. As you Anik is so much busy with his new project, he will accept the challenge if you can help him.

The challenge is to take input two lines and find the multiplication of these numbers.

Anik need help of a programmer to prove himself a master of a language(!). ;)

**Input Format**

*There are****N****tescase, each test case contains two lines always start with character '0' followed by '.', then unknown number****x****and next line contains****y****, each line will always terminated by three dots '...' .*

**Constraints**

x<2^31-1; y<2^31-1;

**Output Format**

Output the multiplication of x,y result = x\*y;

**Sample Input 0**

1

0.1227...

0.517611738...

**Sample Output 0**

635109602526

--------------------------------------------------------------------------------------------------------

02

Mr. Hundred CommentMan was quite busy with his new company. They are building a new product named J.A.R.V.I.S kit. The pre-order calculation days are over. Now, they need to deliver the promised J.A.R.V.I.S kit as soon as possible. In order to deliver the products within the deadline, they need enough intelligent minds. So, CommentMan called me and asked me to initiate a interview. So here I am.

You need to answer only one question in order to pass the first round. Do you have what it takes to be the part of our company?

Just raise number 6 to the power of **n** and get last digit of the number. Yes, of course, n can be rather big, and one cannot find the power using a calculator, but we need people who are able to think, not just follow the instructions.

**Input Format**

The only line of the input contains a single integer — the power in which you need to raise number 6.

**Constraints**

2 ≤ n ≤ 2·10^18

**Output Format**

Output the last digit of 6^n.

**Sample Input 0**

4

**Sample Output 0**

6

**Explanation 0**

6^4 = 1296. So, the last digit is 6.

----------------------------------------------------------------------------

03

Let us define the parity of an integer m as the sum of the bits in binary representation.For example, 11 = 1011 has three 1s in its binary representation so it has parity 3.Given an array of n integers you will have to sort these integers in ascending order of their parity value.If two numbers have equal parity put the largest one first.

**Input Format**

Each test case starts with an integer n denoting the array size.Input is terminated by a line containing a zero.

**Constraints**

**Output Format**

For each set of input out n numbers, sorted according to rules stated avobe.

**Sample Input 0**

10

1

2

3

4

5

6

7

8

9

10

0

**Sample Output 0**

8

4

2

1

10

9

6

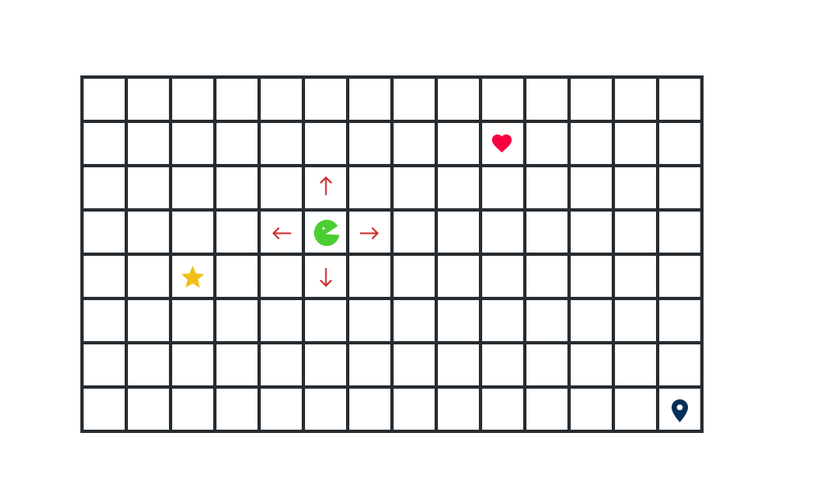
5

3

7

---------------------------------------------------------------------------------------------------------

04

You probably have played Pac-Man many times. But you are not just a gamer anymore, you are a game maker! So this problem would test your game building skills. Given the 2D plane of a Pac-Man game level, your program would show at any given point, how far the collectibles of Pac-Man are.The collectibles are:  
1. Star - gives him +500 pts  
2. Heart - gives him +1 life  
3. Door/Destination - wins the game if he reaches it!

**Input Format**

The first line of input has 2 integers N and M. N denoting the row and M denoting the column of the grid. Next three lines contain 3 pair of integers.  
  
(pos\_x, pos\_y) -> denoting the vertex of the Pac-Man  
(a, b) -> denoting the vertex of Star  
(p, q) -> denoting the vertex of Heart  
(x, y) -> denoting the vertex Door

A (-1, -1) pair means there are no such collectible in this game.

**Constraints**

1<=N, M<=100  
(0, 0) <= (pos\_x, pos\_y), (a, b), (p, q), (x, y) <= (n-1, m-1)

**Output Format**

Output of your program will show the player of the game how far the collectibles are. Print the collectible Name and (after a space) Manhattan Distance from it to Pac-Man because Pac-Man can only move vertically and horizontally, not diagonally. Print "None" (without quotes) if there are no such collectible. See sample I/O for more information.

**Sample Input 0**

10 10

0 0

4 5

-1 -1

9 9

**Sample Output 0**

STAR 9

HEART None

DOOR 18

----------------------------------------------------------------------------------------------

05

As you all know Sheldon has eidetic memory. Which means is an ability to vividly recall images from memory after only a few instances of exposure, with high precision for a brief time after exposure, without using a mnemonic device. But he has a not so smart friend, Leonard. Again Leonard is struggling with Number Theory. So, Sheldon gave him very easy topic to study. Square Free Integers.

In mathematics, a square-free integer is an integer which is divisible by no perfect square other than 1. For example, 10 is square-free but 18 is not, as 18 is divisible by 9 = 3^2. The smallest positive square-free numbers are

1, 2, 3, 5, 6, 7, 10, 11, 13, 14, 15, 17, 19, 21, 22, 23, 26, 29, 30, 31, 33, 34, 35, 37, 38, 39, ...

Now that Leonard have some information regarding square free integer, now he need to practice to be perfect. Sheldon gave him a problem which Leonard must solve in a efficient way to impress the Mr. Know-It-All.

Given a number **n**, find the **n-th** square-free number. A number is square-free if it is not divisible by a perfect square other than 1. As Leonard's girlfriend Penny knows nothing about mathematics you must help him to solve this problem.

**Input Format**

The first line contains a integer N (1 ≤  N ≤ 10^9).

**Constraints**

(1 ≤  N ≤ 10^9).

**Output Format**

Output single integer — answer to the problem.

**Sample Input 0**

20

**Sample Output 0**

31

------------------------------------------------------------------------------------------

06

In the Blue Whale world, blue is considered great color! There are three prime colors in Blue Whale world. Every color has an ID number. White is 1000, Purple is 2000 and Blue is 3000. Other colors are shades of those prime colors! Now given a color in it's Binary form, you have to determine if it's Blue, Almost Blue or 'Eh!'. Because Blue for the win!

Blue : If the color ID exactly matches Blue color's ID  
Almost blue : If the absolute difference from the color and Blue is less then or equal to 500 and the color ID is a prime number.  
Eh! : If it's not even close.

**Input Format**

The only line of input will contain a binary string b.

**Constraints**

The binary string will be less then or equal to 14 bits.

**Output Format**

Print "Blue" (without quotes) or "Almost Blue" (without quotes) or "Eh!" according to the data you analyzed.

**Sample Input 0**

10001

**Sample Output 0**

Eh!

------------------------------------------------------------------------

07

Do you know **Tanvir Sojal**?

**Tanvir Sojal** likes math. He used to pass his times by solving difficult mathematical terms.

One day when he was going to university campus he saw a billboard of his favourite actress "**INNA**"(INNA is Romanian singer and model). He thinking about his favourite actress name. He tried to spell her name reversely. Then it became "ANNI" and he became frustrated. Then he looking for the word which is same after speeling reversely as like the main word. Such as- APPA -> APPA, MADAM -> MADAM, AMMA-> AMMA, 123454321 ->123454321 and so on.

He named this kind of word or number "Parotum" (Parotum is a japanies word).

But finding a Parotum word or a number is not an easy work from the huge number of wordlist and numbers of this word. So he is looking for a bad guy but good at programming, who can write a program that will help him to check is it a "**Parotum**" or "**NOT**" and how much digit or letter occured on this word or number more than one time.

**Input Format**

Input will contain T, N , S (Where T and N are intiger and S is a string). T denotes number of testcase, N denotes size of the Number or word and S input the word or Number.

**Constraints**

0

N<=10000

**Output Format**

Output: If this is a Parotum

Case X: Parotum , P

Where X is the number of test case and P is the number of digit or letter occures more than one times. else

Case X: No Parotum

**Sample Input 0**

3

4 ASAS

2 AB

3 POP

**Sample Output 0**

Case 1: No Parotum

Case 2: No Parotum

Case 3: Parotum , 1

**Explanation 0**

Case - 1 : ASAS -> SASA Not Parotum Case - 2 : AB -> BA Not Parotum Case - 3 : POP -> POP , Parotum , P occured 2 times

-----------------------------------------------------------------------------------------------------

08

Once a day a poor man got an extraordinary goose. The goose gives him some golden eggs in everyday. One day his little curious son asked him that "How many eggs the goose gave us till today?" The poor man is very weak in counting. Can you help him out..? If you then your task is to calculate how many eggs the goose gave them in Ath day to Bth day [A,B] inclusive. You will be given the number of eggs the goose give in a particular day.

**Input Format**

The first line will consist of two integer n and q where 'n' is the number of day and 'q' is the number of query you have to perform.

2nd line will consist of n integer which indicate the number of eggs the goose give in Ai {Ai, A(i+1), A(i+2)....A(i+n) } day. each element will be separeted by a blank space.

Then Q lines will be given. Each line will consist of two integer p and q indicating pth day to qth day. You have to calculate the total number of eggs the goose given in pth to qth day where [p,q] inclusive.

**Constraints**

1 <= n,q <= 100000 ; 1 <= Ai <= 100000 ; 1 <= p,q <= n

**Output Format**

Output will consist of only 1 line. the total number of eggs the goose given in pth to qth day where [p,q] inclusive.

You have to print: "Case K: result" where K is the case no and result is your output.

NB: Your output have to look like sample output exactly.

**Sample Input 0**

5 3

1 2 3 4 5

1 3

2 4

1 5

**Sample Output 0**

Case 1: 6

Case 2: 9

Case 3: 15

---------------------------------------------------------------------------------------------------

09

Input two intiger A,b. output their AxB. As the result can be large use long long integer.

**Input Format**

Input A,B

**Constraints**

-10^9<=A<=B<=10^9

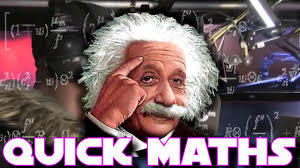
**Output Format**

print (AxB)

------------------------------------------------------------------------------------

10

You know that two plus two is four. You also know that four minus one that's three. Pretty easy, right? Your task is to write a big (!) program that will do quick maths for you!



**Input Format**

Input consists of a single line containing the equation. There will be no space, just operands and operators. The operators will be + (addition) and - (subtraction). And the operands would be integers and there will be no brackets. See the sample for clarification.

**Constraints**

The equation may contain upto 100 characters. All the integers would be less than 10.

**Output Format**

Print the result of the equation, the value could be positive or negative.

**Sample Input 0**

2+2-1

**Sample Output 0**

3

ANS:

int main()

{

int result = 0;

char c[100];

scanf("%[^\n]%\*c", c);

for (int i = 0; i <sizeof(c) / sizeof(c[0]); i++)

{

if (c[i] == '+' || c[i] == '-' || (c[i] >= '0' && c[i] <= '9'))

{

if (c[i - 1] == '+')

{

result = result + (c[i] - '0');

} else if (c[i - 1] == '-')

{

result = result - (c[i] - '0');

} else if (i == 0)

{

result = result + c[i] - '0';

}

}

}

printf("%d", result);

return 0;

}

-------------------------------------------------------------------------------------------------------------------------------------