CodeChef Discussion

ABX01 - EDITORIAL

PROBLEM LINK:

4 Practice
Contest

Setter : Mohammad Salik
4 **Tester :** Hasan Jaddouh
Editorialist : Mohammad Salik

Difficulty

Simple/Easy

Prerequisites

Fast modular exponentiation, Observation skills

Problem

You are given two integers A and N. The task is to compute $F(A^N)$ where $F(X)$ is a function which results in a non-negative single digit integer obtained by a process of summing digits,and on each iteration using the result from the previous iteration to compute a digit sum untill single digit number is reached.

Explanation

For Subtask 1

We can simply find A^N as it doesn't exceed 10^{15} and then evaluate $F(A^N)$ by the recursive process mentioned in the definition of this function.Note that we can evaluate $F(A^N)$ in less than 5 iterations.

For Subtask 2 and 3

Let us first observe the function F first.What will be $F(X)$ for a single digit integer X ? It will simply be the number itself.Or we can say for a single digit integer X ,$F(X) = X\%9$ for $X! = 9$ and $F(X) = 9$ for $X = 9$.Combining these two we can write $F(X) = X\%9 + 9 * (X\%9 == 0)$for a single digit integer X where $X\%9 == 0$ will return 1 only when X is 9. Now what will be $F(X)$ for a two digit integer X. For a two digit integer X which is a multiple of 9 observe that $F(X) = 9$ and for X not a multiple of 9 , $F(X) = X\%9$. Generalising this we can define $F(X)$ as follows : $F(X) = X\%9 + 9 * (X\%9 == 0)$for any non negative integer X where $(X\%9 == 0)$ will return 1 only when X is a multiple of 9.

Now lets try to find out $F(X * X)$. Consider two Cases :

- When X is a multiple of 9 i.e $F(X) = 9$ then $X * X$is also a multiple of 9 and consequently $F(X * X) = 9$. Also see here that $F(F(X) * F(X)) = F(9 * 9) = F(81) = 9$.Hence when X is a multiple of 9 then $F(X * X) = F(F(X) * F(X))$.
- When X is not a multiple of 9 then $F(X) = X\%9$. Hence, we have $F(F(X) * F(X)) = F((X\%9) * (X\%9)) = ((X\%9) * (X\%9))\%9 = (X * X)\%9 = F(X * X)$.Hence in this case also $F(X * X) = F(F(X) * F(X))$.

So lets generalise this. $F(X^2) = F(F(X) * F(X))$. Similarly, $F(X^4) = F((F(F(X) * F(X)))^2)$.

Now for **SUBTASK #2** where $N <= 100$ we can evaluate $F(A^N)$ by iterating from 1 to N and taking F at each step while multiplying.The following code evaluates this in N steps :

```
ll Res=1;
for(int i=1;i<=N;i++)
{
        Res=Res*A;
        Res=F(Res);
}
```

For **Subtask #3** we have $N <= 10^{18}$ and hence we cannot take N steps as it will timeout. we can write a function similar to a fast modular exponentiation to evaluate $F(A^N)$ which evaluates this in $logN$ steps.The following code evaluates this :

```
int solve(long long A,long long N)
{
   long long res=1;
   while(N)
   {
        if(N%2==1)
        {
            res=res*F(A);
            res=F(res);
        }
        A=F(F(A)*F(A));
```

Follow this question
By Email:
You are not subscribed to this questio
subscribe me
(you can adjust your notification settir on your profile)
By RSS:
Answers
Answers and Comments

Question tags:

editorial **×12,622**
simple **×845**
ltime55 **×139**
observations **×61**
fast-expo **×54**
abx01 **×32**

question asked: **29 Dec '17, 18:36**
question was seen: **2,969 times**
last updated: **9 hours ago**

Related questions

NOKIA - Editorial
FEST03 - Editorial
CF221 - Editorial
CDQU5 - Editorial
PLZLYKME - Editorial
MIKE2 - Editorial
PROB - Editorial
CAKEDOOM - Editorial
TADELIVE - Editorial
RIT01 - Editorial

```
        N/=2;
    }
    return res;
}
```

The function $F(A)$ for an integer A can be computed in $O(1)$ easily as we have already defined this function.

Time Complexity

$O(logN)$ per testcase

Space Complexity

$O(1)$

AUTHOR'S AND TESTER'S SOLUTIONS:

Author's solution can be found here.
Tester's solution can be found here.

ltime55 simple observations editorial abx01 fast-expo

This question is marked "community wiki".

edited **30 Dec '17, 23:07**
admin ♦♦
[17.5k]●347●487●515

asked **29 Dec '17, 18:36**
4★ abx_2109
[234]●1●10
accept rate: 0%

16 Answers: oldest answers newest answers **popular answers**

5

As some wonder why there is a magic modulo 9, here is a short explanation:

Any number can be written as a sum of power of 10 and you can write it like this:

1543 = 1 * (999+1) + 5 * (99+1) + 4 * (9+1) + 3

If you rearrange:

1543 = (1 * 999 + 5 * 99 + 4 * 9) + (1 + 5 + 4 + 3)

And then you understand why mod 9 is the magic trick.

More details here: http://www.flyingcoloursmaths.co.uk/a-neat-number-trick-digital-roots-and-modulo-9-arithmetic/

link | award points edited **2 days ago** answered **2 days ago**
3★ sideralis
[65]●2
accept rate: 100%

4

When you realize that $F(A) \equiv A \mod 9$ all you need to do is compute $A^N \mod 9$.

Solution.

And that's also what the tester's solution does.

link | award points edited **31 Dec '17, 01:19** answered **31 Dec '17, 00:48**
3★ eugalt
[148]●4
accept rate: 7%

2

I did it in a different way. I took the sum of digits according to the question given in a query. And then found out the pattern in which result is repeating. Finally, printed the result according to the condition satisfied by the exponential term given in the query.

Here is my solution.

link | award points answered **31 Dec '17, 14:39**
3★ satyajitd_nit
[21]●3
accept rate: 0%

1

i have done this question in O(1) link:-https://www.codechef.com/viewsolution/16717619

link | award points answered **31 Dec '17, 00:28**
3★ bhpra
[88]●6
accept rate: 6%

3 Are you sure it's O(1)? ;)

3★ eugalt (31 Dec '17, 04:27)

after reading the comment of @pk301 i realized that it is O(no of digits) i didn't do a proper analysis sorry saw you solution it is actually O(1) well done

3★ bhpra (yesterday)

O(no of digits) is O(log N).

3★ eugalt (yesterday)

And here is O(1) solution.

1 link | award points edited **2 days ago**

answered **31 Dec '17, 22:08**
3★ eugalt
[148]●4
accept rate: 7%

well done @eugalt if any one didn't get % 9 part the can look the comment of @sideralis nice explanation both of you @eugalt your code is very well represented.

3★ bhpra (yesterday)

1 I was trying to use recursive version of fast exponentiation algorithm but getting TLE on 2nd and 3rd subtask. Can anyone help me regarding same? Please upvote this answer so that I can ask queries :)

https://www.codechef.com/viewsolution/16740009

link | award points

answered **yesterday**
count_me_in
[11]●1
accept rate: 0%

0 Can anyone tell me the mistake in this solution? It passed for the last test case but failed in the first two. I am not getting the mistake.

https://www.codechef.com/viewsolution/16722563

link | award points

answered **31 Dec '17, 01:42**

2★ mittal_ash7
[1]●1
accept rate: 0%

Consider this TC:

2

18 0

18 1

Expected o/p:

1

9

Your code gives:

18

18

Hope this helps!

3★ ayan_nitd (31 Dec '17, 16:38)

i got the mistake. cout<<y<<endl; instead of cout<<a<<endl;. BTW i have not considered n=0 as per constrained. Thanks a lot.

2★ mittal_ash7 (01 Jan, 04:52)

0 My this code is passing for 70 pts and getting WA for first two subtasks. I don't know why it is happening atleast in first subtask because i have matched my output of all the possible test cases for first subtask with ouput of an AC code. If anyone could please help. @abx_2109

link | award points

answered **31 Dec '17, 08:31**

3★ luvk1412
[274]●6
accept rate: 12%

1 For $A = 18, N = 0$ and $A = 18, N = 1$ your code gives 9 and 0 respectively which should be 1 and 9.

3★ ayan_nitd (31 Dec '17, 16:42)

thanks a lot @ayan_nitd for second test case. I found the mistake(18,0 is not a valid testcase)

3★ luvk1412 (yesterday)

0 I have solved this in O(no_of_digits_in_N) but not sure whether my approach is fully correct or not. I have observed that after certain number of steps there will be a cycle and thus i just sum the digits of N untill it becomes single digit and then just print the answer for that single digit for that power.

my solution : https://www.codechef.com/viewsolution/16708827

I don't know whether it fully correct or not and if it is and someone has some mathematical stuff regarding it's proof then please share :)

link | award points

answered **31 Dec '17, 08:44**
2★ pk301
[295]●7
accept rate: 15%

0 This is my solution.

link | award points

answered **31 Dec '17, 12:09**

2★ gautamcse27
[37]●4
accept rate: 0%

DONE THIS USING OBSERVATION https://www.codechef.com/viewsolution/16726993

0

link | award points

answered **31 Dec '17, 19:17**

2★ paranoia
[1]
accept rate: 0%

How did u figured out that F(X)=X%9 for X!=9

0

link | award points

answered **2 days ago**

2★ jk9873
[1]
accept rate: 0%

@abx_2109 u can do this question more efficiently when each time for updating res u should go for checking

0

if(res%9==0) res=9;else res%=9;

rather than again making an recursive function as given in definition

And similarly updating A with

```
A = ((A%M)*(A%M))%M;
```

```
if(A%9==0) A=9;else A%=9;
```

link | award points

answered **2 days ago**

3★ coder_ishmeet
[21]●4
accept rate: 0%

Problem Link **Contest Practise**

0

Setter : Mohammad Salik

Tester : Hasan Jaddouh

Editorialist : Ashwany Aggarwal

Difficulty: Easy

Prerequisites: Mathematics,Observation skills

Problem You are given two integers A and N. The task is to compute F(A^N) where F(X) is a function which results in a non-negative single digit integer obtained by a process of summing digitsuntil single digit number is reached.

Explanation:

STEP 1: First of all we find the sum of integers of the given integer A until it reaches to a single integer (say y), using function :

```
int digsum(unsigned long long p)
{ unsigned long long x,sum=0;
     while(p>0)
               {
                 while(p != 0)
                 {
                      x = p%10;
                      sum = sum+x;
                      p=p/10;
                 }
                 if(sum > 9)
                 {
                         p = sum;
                        sum = 0;
                 }
               }
               return sum;
}
```

Now, by the property of exponents that sum of digits repeat after every 6th exponent.i.e, digsum(A^N)= digsum(A^(N%6)), (except for A=3 or A=6 , where A^N= A for N=1 and A^N=9 for N>1).

For eg. 2^1=2, 2^7=128=1+2+8=11=1+1=2

For eg: 3^1=3; 3^4=81=8+1=9, 6^3=216=2+1+6=9 (you can check for any N).

STEP 2:

Now we have y (the single digit sum for A). Now We have 2 cases either y=(3 or 6) or y= else than 3 or 6.

1st case: If y= 3 or 6 or 9

1: If N>1 output 9

2: If N=1 output Y

2nd case: Y is not 3 or 6 or 9.(I have taken 9 also because for 9 answer is always 9)

Then take remainder of the given large N when divided by 6 (say z).(The main LOGIC) Now single digit raised to power maximum 5 will be 3 digit in worst case this reduces the complexity of problem.

Step 3: Find y^z and apply the same function. You will get the required answer.

Here is my solution: https://www.codechef.com/viewsolution/16728889

link | award points edited **2 days ago** answered **2 days ago**

2★ mittal_ash7
[1]●1
accept rate: 0%

0

We can use "set stl " also

link | award points answered **yesterday**

3★ mr__cyborg
[0]
accept rate: 0%

0

Superfast :)

link | award points edited **9 hours ago** answered **11 hours ago**

3★ eugalt
[148]●4
accept rate: 7%

Your answer

[hide preview] community wiki:

Preview

Type the text

Privacy & Terms

Answer the question