

TCS Code Vita _ Past Questions VI

Problem1: Code And Sum Love Problem

Code And Sum Love

Given a series of numbers find the last two digits of the sum of their magic pairs.

Note:

Magic pair of a number is the last two digits of 2 to the power of the number.

Constraints

$1 \leq N \leq 10^7$ $0 \leq x \leq 10^{18}$

Input Format

First line contains an integer N

Second line will contain N numbers delimited by space

Output

Number that is the output of the given code by taking inputs as specified above

Sample Input 1:

4

8 6 7 4

Sample Output 1: 64

Explanation:

Magic pair of 8 = 56

Magic pair of 6 = 64

Magic pair of 7 = 28

Magic pair of 4 = 16

Sum is 164

Last two digits 64

Sample Input 2:

3

1 2 3

Sample Output 2: 14

SNo	Input	Output
1	2 5 6	96
2	3 1 2 3	14
3	7 2 4 6 8 10 12 14	44
4	5 10 20 30 5 2	60
5	4 8 6 7 4	64

```

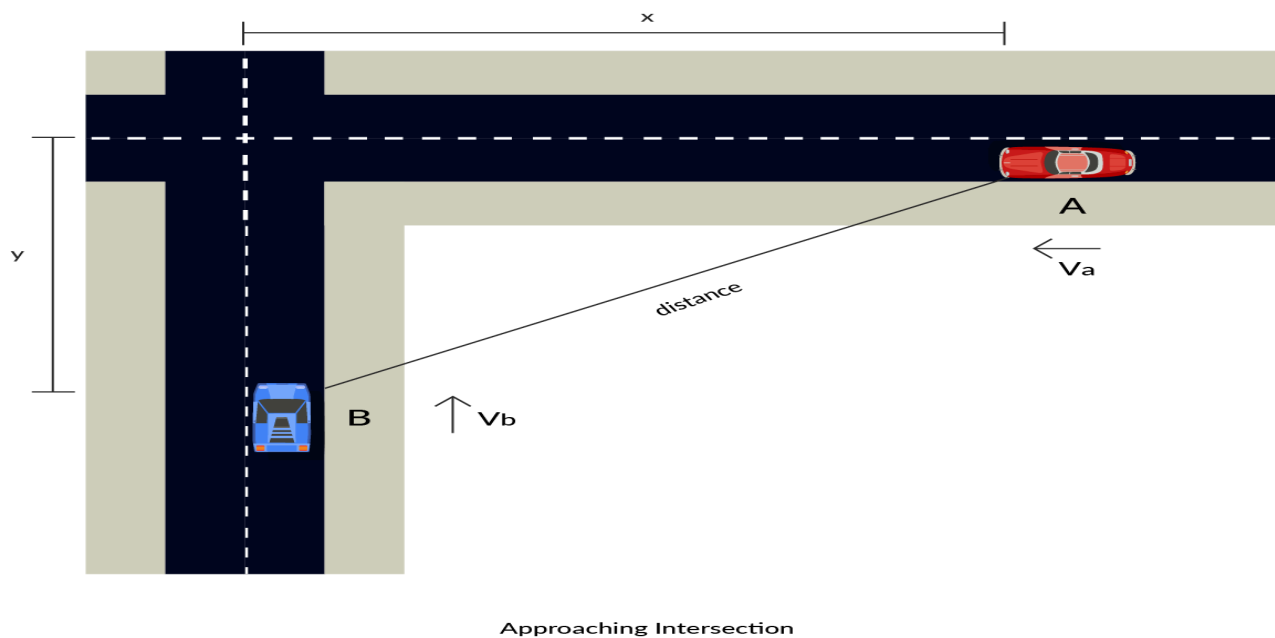
c=[]
n=int(input())
p=list(map(int,input().split()))
for i in range(len(p)):
    l=2**p[i]
    if(l<=99):
        c.append(l)
    else:
        l=l%100
        c.append(l)

o=sum(c)
if(o<=99):
    print(o)
else:
    s=o%100
    print(s)

```

Problem 2: Minimum Distance Problem

Two riders A and B are travelling on a highway towards each other on two roads that intersect at right angle at speeds V_A meters/second and V_B meters/second. A is at a distance of 'x' meters and B is at a distance of 'y' meters from the intersection. Calculate the minimum distance between these two riders that is possible.



Input Format:

First line contains the distance of Rider A from intersection denoted by x

Second line contains the distance of Rider B from intersection denoted by y

Third line contains the Velocity of Rider A denoted by V_A

Fourth line contains the Velocity of Rider B denoted by V_B

Output Format:

Print the minimum distance between these two riders, if minimum distance is non-zero.
If minimum distance is zero, print it as 0.0

Constraints:

$x > 0$

$$y > 0$$

$$V_A > 0$$

$$V_B > 0$$

Calculation and printing of output should be done upto 11-precision

Sample Input 1:

100

100

10

10

Output 1

0.0

Sample Input 2:

500

300

20

14

Sample Output 2:

41.18252056395

Sample Input 3:

05

20

30

Sample Output 3:

Invalid Input

SNo	Input	Output
1	100 100 30 40	22.36067977500
2	0 50 -20 30	Invalid Input
3	500 300 20 14	41.18252056395
4	100 100 10 10	0.0

```

import math
x=int(input())
y=int(input())
va=int(input())
vb=int(input())
if(x<0 or y<0 or va<0 or vb<0):
    print("Invalid Input")
else:
    mint=math.sqrt(x*x+y*y)
    while(x>=0 or y>=0):
        x-=va
        y-=vb
        d=math.sqrt(x*x+y*y)
        if(d<mint):
            mint=d
    if(mint==0.0):
        print("0.0")
    else:
        print("{:.11f}".format(mint))

```

Problem3: Credit and Risk Calculator Problem

Money Bank is an investment bank. It gives money to the companies for their operation that approach it. The bank's officers have to calculate the maximum amount that the company can be given, based on the company's market value and its market rating

published by Global Rating Companies. The maximum amount will change on a daily basis as per the change in the company's shares, its market value and its change in rating. Based on the maximum amount allocated, the bank internally calculates the amount that the company can use on a particular day. The maximum amount that the bank will give at any point is 50% of the company's value, which will decrease as per the change in the company's rating.

Example:

Company ABC has 25734 shares in the market. The current value of the share is 77 INR. The change in the Share Value from yesterday is +10(or 10) Today's rating of ABC is 87 (out of 100). The change in rating from yesterday is +2 (or 2). The credit and risk calculator will work in the following way - Similarly, company XYZ has 30000 shares in the market. The current value of the share is 90 INR. The change in the Share Value from yesterday is -15. Today's rating of XYZ is 83 (out of 100). The change in rating from yesterday is -4. The credit and risk calculator will work in the following way-

No. Of Shares (N)	Share Value (SV)	Change in Share Value (CSV)	Previous Share Value (SV') = SV - CSV	Company Value (CV) = N * min(SV, SV')	Company Rating (R)	Change in Rating (CR)	Previous Rating (R') = R - CR	PMax Credit Company is Eligible for (CE) = CV * (50%)	PCredit Allotted (CA) = CE * (min(R', R)/100)
25734	77	10	67.00	$= (N * 67)$ $= 25734 * 67$ $= 1724178.00$	87	2	85.00	$= 1724178.00 * 0.5$ $= 862089.00$	$= (CE * 85 / 100)$ $= 862089 * 0.85$ $= 732775.65$
30000	90	-15	105.00	2700000.0	83	-4	87	1350000.00	1120500.00

Input Format:(Headings in bold are input in above table)

First line contains total number of test cases T

Each test case comprises of

- First line contains number of shares **N**
- Second line contains current share value **SV**
- Third line contains change in share value **CSV**

- Fourth line contains current rating **R**
- Fifth line contains change in rating **CR**

Output Format: (Headings in blue are output in above table)

In case of Valid inputs, print the following per line

Previous Share Value **SV'**

Previous Rating **R'**

Company Value **CV**

Maximum Credit **CE**

Credit Allotted **CA**

OR

Print "Invalid Input" in case of invalid input(s)

Constraints:

Highest Priority Constraint

- Share Value (SV and SV') can never be negative

Input Constraints:

- $1 \leq T \leq 100$
- $20000 \leq N \leq 100000000$
- $20.00 \leq SV \leq 10000.00$ (**Note:-** This is a numerical constraint only)
- $-2000.00 \leq CSV \leq 2000.00$ (**Note:-** This is a numerical constraint only)
- $0.01 \leq R \leq 99.99$ (**Note:-** This is a numerical constraint only)
- $-10.00 \leq CR \leq 10.00$ (**Note:-** This is a numerical constraint only)
- Calculation should be done upto 11-digit precision

The Change in Share Value (CSV) and Change in Rating (CR) are dependent as below:

1. If CSV is Positive, CR can be Positive or Negative.
2. If CSV is Negative, CR cannot be Positive.

Output Constraints:

The values of CV, CE and CA are to be rounded to 2 Decimal Places, always to the Ceiling or Upper Value

The range of the calculated values are-

- $20.00 \leq SV' \leq 10000.00$
- $0.01 \leq R' \leq 99.99$
- All output values should be printed upto 2 decimal places.

The output will be invalid in the following cases:

The input ranges or conditions are not satisfied.

The ranges of calculated values are not satisfied.

The relation between CSV and CR is not satisfied.

Sample Input 1:

25000
35
5
85
6

Sample Output 1:

30.00
79.00
750000.00
375000.00
296250.00

Sample Input 2:

20000
19
2
60
3

Sample Output 2:

Invalid Input

SNo	Input	Output
1	750000 20 -500 99.99 -10	Invalid Input

2	25000 35 5 85 6	30.00 79.00 750000.00 375000.00 296250.00
3	750000 20 -500 94.99 2	Invalid Input
4	750000 20 -500 94.99 -5	520.00 99.99 15000000.00 7500000.00 7124250.00
5	30000 90 -15 83 -4	105.00 87.00 2700000.00 1350000.00 1120500.00
6	22000 30 -5 46 6	Invalid Input
7	20000 20 3 60 3	Invalid Input
8	20000 19 2 60 3	Invalid Input
9	5000000 5000 1500 50 7	3500.00 43.00 17500000000.00 8750000000.00 3762500000.00

Partial Solution

```

x1=float(input())#number of shares
x2=float(input())#share values sv
x3=float(input())#change in sv
x4=float(input())#rating r
x5=float(input())#change in r
svc=x2-x3#previous share value
rc=x4-x5#previous rating
cv=x1*min(x2,svc)#pcompany value
ce=cv*(0.5)#company eligibility
ca=ce*((min(rc,x4))/100)#pc credit
if(((x1<=(10**7)) and (x1>=20000)) and ((x2>=20.00) and (x2<=10000)) and
((x3>=-2000) and (x3<=2000)) and ((x4>=0.01) and (x4<=99.99)) and ((x5>=-10)
and (x5<=10)) and ((svc>=20) and (svc<=10000)) and ((rc>=0.01) and
(rc<=99.99))):
    print("%.2f"%svc)
    print("%.2f"%rc)
    print("%.2f"%cv)
    print("%.2f"%ce)
    print("%.2f"%ca)
else:
    print("Invalid Input")

```

Problem4: Online Communities - Even Groups Problem

In a social network, online communities refer to the group of people with an interest towards the same topic. People connect with each other in a social network. A connection between Person **I** and Person **J** is represented as **C I J**. When two persons belonging to different communities connect, the net effect is merger of both communities which I and J belonged to.

We are only interested in finding out the communities with the member count being an even number. Your task is to find out those communities.

Input Format:

Input will consist of three parts,

1. The total number of people on the social network (**N**)
2. Queries
 - **C I J**, connect I and J
 - **Q 0 0**, print the number of communities with even member-count

-1 will represent end of input.

Output Format:

For each query Q, output the number of communities with even member-count

Constraints:

$1 \leq N \leq 10^6$

$1 \leq I, J \leq N$

Sample Input 1:

```
5
Q 0 0
C 1 2
Q 0 0
C 2 3
Q 0 0
C 4 5
Q 0 0
-1
```

Sample Output 1:

```
0
1
0
1
```

Sample Input 2:

```
3
C 1 2
Q 0 0
C 2 3
Q 0 0
C 1 3
Q 0 0
-1
```

Sample Output 2:

```
1
0
```

0

Explanation:

For first query there are no members in any of the groups hence answer is 0.

After C 1 2, there is a group (let's take it as G1) with 1 and 2 as members hence total count at this moment is 1.

After C 2 3 total members in G1 will become {1, 2, 3} hence there are no groups with even count.

After C 4 5 there formed a new group G2 with {4, 5} as members, hence the total groups with even count is 1.

SNo	Input	Output
1	5 Q 0 0 C 1 2 Q 0 0 C 2 3 Q 0 0 C 4 5 Q 0 0 -1	0 1 1
2	3 C 1 2 Q 0 0 C 2 3 Q 0 0 C 1 3 Q 0 0 -1	1 0
3	7 Q 0 0 C 1 2 Q 0 0 C 2 3 Q 0 0 C 4 5 Q 0 0 C 3 5 Q 0 0 C 6 7 Q 0 0 -1	 1 1 1

4	2 C 0 0 Q 0 0 C 1 2 Q 0 0 -1	0 1
5	1 Q 0 0 C 1 1 -1	0
6	6 Q 0 0 C 1 2 Q 0 0 C 3 4 Q 0 0 C 5 6 Q 0 0 -1	0 1 2 3

Problem5: Digital Time Problem

Form the maximum possible time in the HH:MM:SS format using any six of nine given single digits (not necessarily distinct)

Given a set of nine single (not necessarily distinct) digits, say 0, 0, 1, 3, 4, 6, 7, 8, 9, it is possible to form many distinct times in a 24 hour time format HH:MM:SS, such as

17:36:40 or 10:30:41 by using each of the digits only once.

Find the maximum possible valid time (00:00:01 to 24:00:00) that can be formed using some six of the nine digits exactly once.

In this case, it is 19:48:37.

Input Format:

A line consisting of a sequence of 9 (not necessarily distinct) single digits (any of 0-9) separated by commas. The sequence will be non-decreasing

Output Format:

The maximum possible time in a 24 hour clock (00:00:01 to 24:00:00) in a HH:MM:SS form that can be formed by using some

six of the nine given digits (in any order) precisely once each. If no combination of any six digits will form a valid time, the output should be the word Impossible

Sample Input 1:

0,0,1,1,3,5,6,7,7

Sample Output 1:

17:57:36

Explanation:

The maximum valid time in a 24 hour clock that can be formed using some six of the 9 digits precisely once is 17:57:36

Sample Input 2:

3,3,3,3,3,3,3,3,3

Sample Output 2:

Impossible

Explanation:

No set of six digits from the input may be used to form a valid time.

SNo	Input	Output
1	9,6,9,8,4,2,5,9,2	22:59:49
2	9,9,9,9,9,0,0,0,0	09:09:09
3	0,4,0,1,0,0,2,0,0	24:00:00
4	1,7,8,6,6,6,6,5,9	Impossible
5	0,0,0,0,0,0,0,0,0	Impossible
6	3,3,3,3,3,3,3,3,3	Impossible
7	0,3,3,9,3,3,3,9,3	09:39:33
8	0,0,0,4,0,0,2,0,0	24:00:00
9	2,0,8,0,4,0,6,5,0	24:00:00
10	0,9,5,5,9,9,9,9,9	09:59:59
11	0,0,1,1,3,5,6,7,7	17:57:36

Problem6: Number Game Problem

Aman and Jasbir are very intelligent guys of their batch. Today they are playing a game "Game of Numbers".

Description of game:

1. There are N numbers placed on a table.
2. Since two players are playing the game, they will make their moves alternatively.
3. In one move a player can perform the following operation.
 - a. A player will choose a number from the table and will replace that number with one of its divisor. For example, 6 can be replaced with 1, 2, or 3 (since these are the divisors of 6). Similarly, 12 can be replaced with 1, 2, 3, 4 or 6.
 - b. It is mandatory that the player has to replace the number.
 - c. A player cannot put back the same number on table.
 - d. As 1 does not have any divisor other than itself, a player cannot replace 1 with any other number. So soon a situation will arise when there will be

only all 1s on the table. In that situation the player will not be able to make any move. The player who will not be able to make the move, loses.

4. Both the players are masters of this game. So they will play the game optimally.
5. Aman will make the first move of the game.

You will be given N integers that are on the table. You have to predict, who will win the game - Aman or Jasbir.

Input Format:

First line contains information about the numbers present of the table, denoted by **N**

Second line contains N integers delimited by space - A1 A2 A3 A4 AN

Output Format:

Print the name of the player who will win the game in upper case i.e. AMAN or JASBIR.

Constraints:

$1 \leq N \leq 100000$

$0 < A_i \leq 10000$

A player cannot replace more than 1 number in one move.

Players move alternate and a player cannot *pass* or make *no move*.

A number cannot be replaced itself, i.e. 6 can be replaced with 1, 2 or 3 only , not with 6.

Aman always makes the first move.

Sample Input 1:

2

18 6

Sample Output 1:

AMAN

Sample Input 2:

4

24 45 45 24

Sample Output 2:

JASBIR

Explanation for Input 1

There are 2 numbers present on the table. Since every move is the optimal move, Aman will replace 18 with 6. Now (6, 6) will be left on the table. Now, whatever move Jasbir will make. Aman will also make the same move. So whenever Jasbir will replace any number with 1. In the next move Aman will also replace the other number with 1. So both the numbers will be one after Aman's move. So Jasbir will lose the game.

Initial configuration - (18, 6)
 AFTER Aman's MOVE - (6, 6)
 AFTER Jasbir's MOVE - (6, 3)
 AFTER Aman's MOVE - (3, 3)
 AFTER Jasbir's MOVE - (1, 3)
 AFTER Aman's MOVE - (1, 1)

Now, Jasbir WILL NOT BE ABLE TO MAKE ANY MOVE. So, **Aman** will WIN.

SNo	Input	Output
1	3 6 8 14	JASBIR
2	3 4 18 20	AMAN
3	1 6	AMAN
4	2 18 6	AMAN
5	4 24 45 45 24	JASBIR
6	6 1 5 6 8 25 19	JASBIR

```
#include<stdio.h>
int main()
{
int i,a,sum=0, arr[100000];
scanf("%d",&a);
for(i=0;i<a;i++)
scanf("%d",&arr[i]);
for(i=0;i<a;i++)
sum=sum^arr[i];
if(sum==0)
printf("JASBIR");
else
```

```
printf("AMAN");
```

```
    return 0;  
}
```