

Lift queries

There are 7 floors in BH3 and only 2 lifts. Initially Lift A is at the ground floor and Lift B at the top floor. Whenever someone calls the lift from N th floor, the lift closest to that floor comes to pick him up. If both the lifts are at equidistant from the N th floor, then the lift from the lower floor comes up.

INPUT

- First line contains a integer T denoting the number of test cases.
- Next T lines contains a single integer N denoting the floor from which lift is called.

OUTPUT

Output T lines containing one character "A" if the first lift goes to N th floor or "B" for the second lift.

CONSTRAINTS

- $0 \leq N \leq 7$
- $1 \leq T \leq 100000$

SAMPLE INPUT

- 2
- 3
- 5

SAMPLE OUTPUT

- A
- A

In [1]:

```
A=0
B=7
N=int(input())
for x in range(0,N):
    Floor=int(input())
    if (Floor-A)<=(B-Floor):
        A=Floor
        print('A')
    else:
        B=Floor
        print("B")
```

```
2
3
A
5
A
```

In []:

Life, the Universe, and Everything

Your program is to use the brute-force approach in order to find the Answer to Life, the Universe, and Everything. More precisely... rewrite small numbers from input to output. Stop processing input after reading in the number 42. All numbers at input are integers of one or two digits.

SAMPLE INPUT

- 1
- 2
- 88
- 42
- 99

SAMPLE OUTPUT

- 1
- 2
- 88

In [2]:

```
N=int(input())  
  
while(N!=42):  
  
    print(N)  
    N=int(input())  
    if N==42:  
        break
```

```
1  
1  
2  
2  
88  
88  
42
```

In []:

Book of Potion making

Harry wants to find Voldemort's potion making book but he is confused about how to get it.

The book has a special ISBN(International Standard Book Number) which is unique numeric book identifier only for Voldemort's book printed on it. The ISBN is based upon a 10-digit code. The ISBN is valid if: $1 \times \text{digit1} + 2 \times \text{digit2} + 3 \times \text{digit3} + 4 \times \text{digit4} + 5 \times \text{digit5} + 6 \times \text{digit6} + 7 \times \text{digit7} + 8 \times \text{digit8} + 9 \times \text{digit9} + 10 \times \text{digit10}$ is divisible by 11. Help Harry to find the book!

Input:

Input the ISBN code as a 10-digit integer.

Output:

- If the ISBN is not a 10-digit integer, output the message "Illegal ISBN" and terminate the program.
- If the number is 10-digit, extract the digits of the number and compute the sum as explained in the

Explanation.

If the sum is divisible by 11, output the message, "Legal ISBN". If the sum is not divisible by 11, output the message, "Illegal ISBN".

SAMPLE INPUT

1401601499

SAMPLE OUTPUT

Legal ISBN

Explanation

For an ISBN 1401601499

$\text{Sum} = 1 \times 1 + 2 \times 4 + 3 \times 0 + 4 \times 1 + 5 \times 6 + 6 \times 0 + 7 \times 1 + 8 \times 4 + 9 \times 9 + 10 \times 9 = 253$ which is divisible by 11.

In [3]:

```
Res=input()
mul=1;sum1=0
if len(Res)==10:
    for x in Res:
        sum1=sum1+(int(x))*mul
        mul+=1
    if sum1%11==0:
        print("Legal ISBN")
    else:
        print("Illegal ISBN")
else:
    print("Illegal ISBN")
```

1401601499

Legal ISBN

In []:

Ladderophilia

Aditya is fond of ladders. Everyday he goes through pictures of ladders online but unfortunately today he ran out of ladder pictures online. Write a program to print “ladder with N steps”, which he can use to get his daily dose of ladder love.

INPUT:

Input contains integer N, the number of steps in the ladder

OUTPUT:

Print the ladder with the gap between two side rails being 3 spaces(“ “).

Check the sample output for format of printing the ladder.

CONSTRAINTS:

$1 \leq N \leq 40$

SAMPLE INPUT

4

SAMPLE OUTPUT

```
*  *
*  *
*****
*  *
*  *
*****
*  *
*  *
*****
*  *
*  *
*****
*  *
*  *
```

In [4]:

```
N=int(input())

for x in range(0,N+1):
    print("*   *")
    print("*   *")
    if x<N:
        print("*****")
```

```
4
*   *
*   *
*****
*   *
*   *
*****
*   *
*   *
*****
*   *
*   *
*****
*   *
*   *
```

In []:

Cipher

Indian army is going to do a surprise attack on one of its enemies country. The President of India, the Supreme Commander of the Indian Army will be sending an alert message to all its commanding centers. As enemy would be monitoring the message, Indian army is going to encrypt(cipher) the message using basic encryption technique. A decoding key 'K' (number) would be sent secretly.

You are assigned to develop a cipher program to encrypt the message. Your cipher must rotate every character in the message by a fixed number making it unreadable by enemies.

Given a single line of string 'S' containing alpha, numeric and symbols, followed by a number ' $0 \leq N \leq 1000$ '. Encrypt and print the resulting string.

Note: The cipher only encrypts Alpha and Numeric. (A-Z, a-z, and 0-9) . All Symbols, such as - , ; %, remain unencrypted.

SAMPLE INPUT

All-convoYs-9-be:Alert1. 4

SAMPLE OUTPUT

Epp-gsrzsCw-3-fi:Epivx5.

In [5]:

```
string=input()
N=int(input())
res=""

N1=N%26
N2=N%10

for x in string:
    if x.islower():
        if ord(x)+N1<=122:
            res+=chr(ord(x)+N1)
        elif ord(x)+N1>122:
            z=ord(x)+N1-122
            res+=chr(z+96)
    elif x.isupper():
        if ord(x)+N1<=90:
            res+=chr(ord(x)+N1)
        elif ord(x)+N1>90:
            z=ord(x)+N1-90
            res+=chr(z+64)
    elif x.isdigit():
        if(ord(x)>=48 and ord(x)<=57):
            if ord(x)+N2<=57:
                res+=chr((ord(x)+N2))
            elif ord(x)+N2>57:
                z=ord(x)+N2-57
                res+=chr(z+47)
    else:
        res+=x
print(res)
```

All-convoYs-9-be:Alert1.

4

Epp-gsrzsCw-3-fi:Epivx5.

In []:

Ali and Helping innocent people

Arpasland has surrounded by attackers. A truck enters the city. The driver claims the load is food and medicine from Iranians. Ali is one of the soldiers in Arpasland. He doubts about the truck, maybe it's from the siege. He knows that a tag is valid if the sum of every two consecutive digits of it is even and its letter is not a vowel. Determine if the tag of the truck is valid or not.

We consider the letters "A","E","I","O","U","Y" to be vowels for this problem.

Input Format

The first line contains a string of length 9. The format is "DDXDDD-DD", where D stands for a digit (non zero) and X is an uppercase english letter.

Output Format

Print "valid" (without quotes) if the tag is valid, print "invalid" otherwise (without quotes)

SAMPLE INPUT

12X345-67

SAMPLE OUTPUT

invalid

In [6]:

```
string=list(input())
y=0
if string[6]=='-' and string[2] not in ['A','E','I','O','U','Y']:
    string.remove(string[6])
    string.remove(string[2])
    sum1=[]
    for x in range(0,len(string)-1):
        if x==4 or x==1:
            continue
        else:
            z=int(string[x])+int(string[x+1])
            sum1.append(z)
    for x in sum1:
        if int(x)%2!=0:
            print("invalid")
            y=0
            break
        else:
            y=1
    if y:
        print("valid")
else:
    print("invalid")
```

12X345-67

invalid