SESHADRI RAO GUDLAVALLERU ENGINEERING COLLEGE
(An Autonomous Institute with Permanent Affiliation to JNTUK, Kakinada)
Seshadri Rao Knowledge Village, Gudlavalleru -- 521356

Department of Computer Science & Engineering

COMITY EVENT – 2

CODING CONTEST

No of Questions 5

Question type: Coding

1. Length of Palindrome

Problem:

You are provided with a number **S** and check whether it is a palindrome or not. If it's nota palindrome, then reverse the number, add it to the original number and check again. You are required to repeat the process until it becomes a palindrome and the number should not exceed 10^7. If it exceeds the limit then return -1,otherwise return length of obtained palindrome.

Input Format

- The first line of the input contains a single integer **T** denoting the number of test cases. The description of **T** test cases follows.
- The next **T** lines contains a number

S.**Constraints**

1 ≤ T ≤ 100 1 ≤ S ≤ 100000

Output Format

- For each test case, if there exists a palindrome, print the length of the obtained palindrome, otherwise print -1 (without the quotes).

Sample Input 0

```
4

67

89

100

11
```

Sample Output 0

```
3

-1

3

2
```

Explanation 0

First line 4 specifies number of test cases. Number 67 is not palindrome. Reverse and add (67+76=143). 143 is not palindrome. Reverse and add (143+341=484). 484 is palindrome. So return length of 484 as 3. In the same manner 100 will be transformedto palindrome and 11 is a palindrome and appropriate lengths were returned. But,89 can't be transformed to palindrome within limit 10^7. So -1 is returned for it.

Solution:

```
m=10**7
for i in range(int(input())):
    a=input()
    if a==a[::-1]:
        print(len(a))
    else:
        t=str(int(a)+int(a[::-1]))
        while t!=t[::-1]:
            t=str(int(t)+int(t[::-1]))
            if int(t)>=m:
                print(-1)
                break
        else:
           print(len(t))
```

2. Recycling Pens

Problem:

You have 'N' empty pens whose refills have been used up. You have 'R' rupees in your pocket. You have two options of procedures that you may do each time.

1) Recycle One empty pen and receive 'K' rupees as a reward.
2) Purchase 1 refill for "C" rupees, combine it with 1 empty pen to create 1 functional pen.

Your job is to locate the highest amount of functional pens that you can acquire.

Example: if you have 'N' = 5 , 'R' = 10 , 'K' = 2 , 'C' = 3. You may recycle one pen and earn 2 rupees as a reward so you will have a total of 12 rupees. Now you can purchase 4 refills and combine it with 4 pens to make it functional. So your answer is 4.

Input Format

• The first line of the input comprises a single integer 'T', signifying the number of test cases. Then the 'T' test case follows.

• The first and the sole line of each test case consists of 4 non-negative integers 'N', 'R', 'K' and 'C', as indicated in the problem statement.

Constraints

1 ≤ T ≤ 10^5

1 ≤ N ≤ 10^9

0 ≤ R ≤ 10^9

1 ≤ K ≤ 10^9

1 ≤ C ≤ 10^9

Output Format

- For each test scenario, output a single integer on a new line, representing the maximum number of useable pens you can get.

Sample Input 0

```
3

10 10 5 5

15 11 3 5

3 20 20 2
```

Sample Output 0

```
6

7

3
```

Explanation 0

In the first test case, you will sell 4 empty pens and you will earn 20 rupees thus your total money will be 10+20 = 30 and from that, you will purchase 6 refills and manufacture 6 useable pens.

In second test scenario you will sell 8 pens and you will earn 8*3*=24 rupees and your total money will be 24+11 = 35 and from that, you will purchase 7 refills and manufacture 7 useable pens.

In this test scenario, you have a lot of money to spend but only three empty pens, so you will purchase three refills for six rupees and turn them into three functioning pens.

Sample Input 1

```
 3

10 10 1 10

5 0 5 5

6 0 4 2
```

Sample Output 1

```
 1

2

4
```

Explanation 1

In the first test case you can purchase 1 refill with the money you have and manufacture 1 functioning pen.

In the second test case you may sell 2 empty pens and gain 10 rupees and from it, you can purchase 2 refills and produce 2 useable pens.

In the third test case you may sell 2 empty pens and gain 8 rupees and from it, you can purchase 4 refills and produce 4 useable pens.

Solution:

```
for i in range(int(input())):
    n,r,k,c=map(int,input().split())
    print(int(min(n,(r+n*k)/(c+k))))
```

3. Beautiful Name

Problem:

Kill and Roj are bored. They want to play a game of names. Roj gives Kill a collection of names so that Kill had to select a beautiful name among them.

A name is considered to be beautiful if it has the least value when the sum of ascii values of letters absent in the name is deducted from the sum of ascii values of letters present in the name.

If a name consists of a capital letter then the complete name has to be translated into small letters.

Input Format

• First line has an Integer N, representing the number of names.

• Next N lines includes a String s each, representing a name.

Constraints

0 ≤ N ≤ 100

Output Format

• print the beautiful name from the collection of names, print -1 if you cannot choose a single beautiful name or if string includes other than characters.

Sample Input 0

```
 3
sai
Chandra
rohit
```

Sample Output 0

```
Chandra
```

Explanation 0

The ascii value of sai = 2530 - 317 = 2213 Chandra is transformed into => chandra = 2126 - 721 = 1405 rohit = 2297 - 550 = 1747 So Chandra is the beautiful name.

Sample Input 1

```
 2
Rajnath
rajnath
```

Sample Output 1

```
 -1
```

Explanation 1

The string Rajnath is transformed to rajnath, now as the two string are similar therefore we cannot decide beautiful name. So Print "-1".

Solution:

```
a=[]

for i in range(int(input())):

    a.append(input())

freq=dict()

lt=sum(list(range(97,123)))

for i in a:

    if i.isalpha():

        t=sum([ord(j) for j in i.lower()])

        freq[i]=lt-(2*t)

    else:

        print(-1)

        break

else:

    t=min(freq.values())

    r=[i for i in freq if freq[i]==t]

    if len(r)==1:

        print(*r)

    else:

        print(-1)
```

4. Password Encryption

Problem:

You want to send your password to your buddy, but you are concerned that third person may steal the password. So, now you have to encrypt the password. Password encryption will be done by executing circular right rotation of special characters in their respective positions N times without modifying the position of alphabets. Special characters include '@','(',')','{','}' only. Password should includes alphabets and special characters only. Otherwise, return -1. If the string doesn't contains any special characters then return the string itself.

Input Format

• First line comprises a string, representing the password.

• Next line has an integral value N, denoting number of shifts necessary.

Constraints

1 ≤ N ≤ 1010

Output Format

• Print the encrypted password if the string comprises only special characters and alphabets. Otherwise, -1.

Sample Input 0

```
@SR@)gudlavalleru{
3
```

Sample Output 0

```
@SR){gudlavalleru@
```

Explanation 0

In above example,it needs 3 circular rotations of special characters

In first shift,it becomes {SR@@gudlavalleru)

In second shift,it becomes)SR{@gudlavalleru@

In third shift,it becomes @SR){gudlavalleru@

Sample Input 1

```
 Hello world
4
```

Sample Output 1

```
 -1
```

Explanation 1

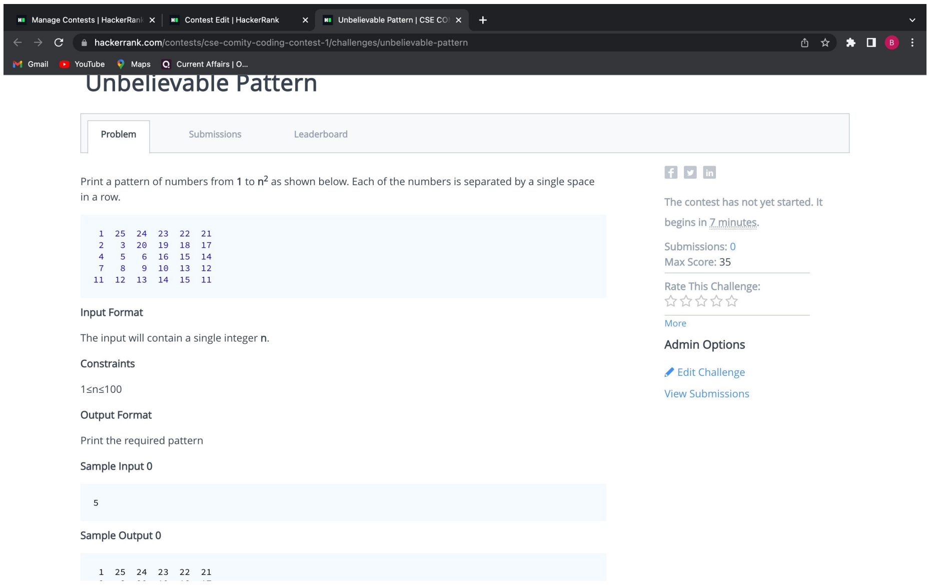
In above example,input string contains space as additional character. So,-1 is the output.

Solution:

```
a=list(input())
n=int(input())
s,p="",[]
for i in range(len(a)):
    if a[i] in ['@','(',')','{','}']:
        s=s+a[i]
        p.append(i)
    elif a[i].isalpha():
        continue
    else:
        print(-1)
        exit()
if len(p) and len(s):
    t=len(s)
    s=s[t-(n%t):]+s[:t-(n%t)]
    j=0
    for i in p:
        a[i]=s[j]
        j=j+1
    print("".join(a))
else:
    print("".join(a))
```

5. Unbelievable Pattern

Problem:


Manage Contests | HackerRank
Contest Edit | HackerRank
Unbelievable Pattern | CSE CO
hackerrank.com/contests/cse-comity-coding-contest-1/challenges/unbelievable-pattern
Gmail
YouTube
Maps
Current Affairs | O...
B
Problem
Submissions
Leaderboard
More
Admin Options
Edit Challenge
View Submissions

Unbelievable Pattern

Print a pattern of numbers from 1 to n^2 as shown below. Each of the numbers is separated by a single space in a row.

```
  1  25  24  23  22  21
  2   3  20  19  18  17
  4   5   6  16  15  14
  7   8   9  10  13  12
 11  12  13  14  15  11
```

Input Format

The input will contain a single integer n.

Constraints

1≤n≤100

Output Format

Print the required pattern

Sample Input 0

```
5
```

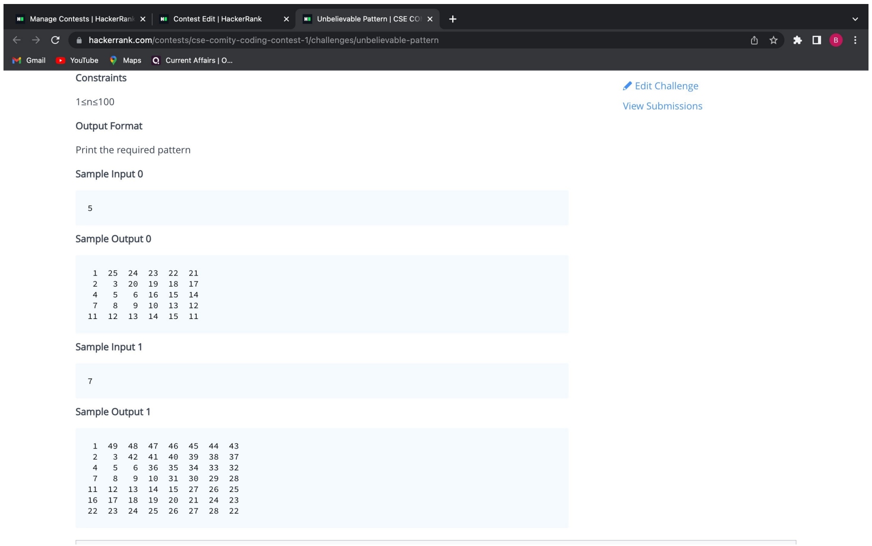
Sample Output 0

```
  1  25  24  23  22  21
```

The contest has not yet started. It begins in 7 minutes.

Submissions: 0
Max Score: 35

Rate This Challenge:
☆☆☆☆☆


Manage Contests | HackerRank
Contest Edit | HackerRank
Unbelievable Pattern | CSE CO
hackerrank.com/contests/cse-comity-coding-contest-1/challenges/unbelievable-pattern
Gmail
YouTube
Maps
Current Affairs | O...
B
Edit Challenge
View Submissions

Constraints

1≤n≤100

Output Format

Print the required pattern

Sample Input 0

```
5
```

Sample Output 0

```
  1  25  24  23  22  21
  2   3  20  19  18  17
  4   5   6  16  15  14
  7   8   9  10  13  12
 11  12  13  14  15  11
```

Sample Input 1

```
7
```

Sample Output 1

```
  1  49  48  47  46  45  44  43
  2   3  42  41  40  39  38  37
  4   5   6  36  35  34  33  32
  7   8   9  10  31  30  29  28
 11  12  13  14  15  27  26  25
 16  17  18  19  20  21  24  23
 22  23  24  25  26  27  28  22
```

Solution:

```
n=int(input())
a=1
b=n*(n+1)
l=len(str(b))
for i in range(n):
    for j in range(n+1):
        if i>=j:
            print((l-len(str(a)))*"  "+str(a),"  ",end="")
            a=a+1
        else:
            print((l-len(str(b)))*"  "+str(b),"  ",end="")
            b=b-1
    print()
```

Dr. T. Srinivasa Rao	**Dr. M. Babu Rao**
(COMITY Advisor)	**(HoD, CSE)**