

WEEK - 1**Problem**

Nitin and Sobhagya were playing a game with coins. If Sobhagya has **more** coins then he is *winning*, otherwise Nitin is winning. **Note** that this means if both Nitin and Sobhagya have the same number of coins, then Nitin is winning.

Initially Nitin has *AA* coins while Sobhagya has *BB* coins. Then Ritik came and gave his *CC* coins to the player who is **not winning** currently, after which Satyarth came and repeated the same process (gave his *DD* coins to the player who is **not winning** currently).

Find the final winner of the game.

Input Format

- The first line of the input contains an integer *TT* - the number of test cases. The test cases then follow.
- The only line of each test case contains four space-separated integers *AA*, *BB*, *CC*, and *DD*.

Output Format

For each test case, output on a single line *N* if Nitin is the final winner of the game, or *S* if Sobhagya is the final winner of the game.

Solution :

```
t = int(input())
for _ in range(t):
    c = list(map(int,input().split()))[:4] # coins
    if(c[0]<c[1]):
        c[0] = c[0]+c[2]
    else:
        c[1] = c[1]+c[2]
    if(c[0]<c[1]):
        c[0] = c[0]+c[3]
    else:
        c[1] = c[1]+c[3]
    if (c[0]==c[1] or c[0]>c[1]):
        print("N")
    else: print("S")
```

Test Cases:**Input:**

```
3
2 3 4 5
3 3 3 3
2 3 1 2
```

Output :

```
S
N
S
```

Result :

CODECHEF Upgrade To Pro rizwan_rockzz Logout

Back to Practice Switch to Old Problems Page Prev Problem Next Problem

Game between friends
 Problem Code: FRGAME Contest Code: CDRV2021 Difficulty Rating: 991 Show Tags

Statement Hints Submissions Solution Ask a Doubt

Find the final winner of the game.

Input Format

- The first line of the input contains an integer T - the number of test cases. The test cases then follow.
- The only line of each test case contains four space-separated integers A, B, C , and D .

Output Format

For each test case, output on a single line N if Nitin is the final winner of the game, or S if Sobhagya is the final winner of the game.

Constraints

- $1 \leq T \leq 1000$
- $0 \leq A, B, C, D \leq 10^6$

PYTH 3

```
1 t = int(input())
2
3 for _ in range(t):
4     c = list(map(int, input().split()))[:4] # coins
5
6     if(c[0] < c[1]):
7         c[0] = c[0] + c[2]
8     else:
9         c[1] = c[1] + c[2]
10
11    if(c[0] < c[1]):
12        c[0] = c[0] + c[3]
13    else:
14        c[1] = c[1] + c[3]
15
16    if (c[0] == c[1] or c[0] < c[1]):
17        print("N")
18    else:
19        print("S")
20
21
```

Statement Hints Submissions Solution Ask a Doubt

Find the final winner of the game.

Input Format

- The first line of the input contains an integer T - the number of test cases. The test cases then follow.
- The only line of each test case contains four space-separated integers A, B, C , and D .

Output Format

For each test case, output on a single line N if Nitin is the final winner of the game, or S if Sobhagya is the final winner of the game.

Constraints

- $1 \leq T \leq 1000$
- $0 \leq A, B, C, D \leq 10^6$

Sample 1:

Status: ✔ Correct Answer Submission ID: 73060121

Time: 0.03s

Sub-Task	Task #	Result (time)
1	0	AC (0.024658)
1	1	AC (0.030645)
Subtask Score: 100.00%		Result - AC
Total Score = 100.00%		

Congratulations on solving the problem. Visit our practice section to solve more interesting problems View another problem

Why only this DS is used? :

This question is all about conditional statements (if, else) successfully executed.

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

WEEK – 2**Problem**

You are given a standard 8 \times 8 chessboard which has exactly 22 rooks on it and no other pieces. The rows are numbered 11 to 88 from bottom to top, and the columns are numbered 11 to 88 from left to right. The cell at the intersection of the ii -th column and jj -th row is denoted $(i,j)(i,j)$.

Given the initial positions of the rooks in the form of coordinates $(X_1,Y_1)(X_1,Y_1)$ and $(X_2,Y_2)(X_2,Y_2)$, you need to tell whether the 22 rooks currently attack each other or not. Assume, each square can contain at most one piece.

Rooks can only travel in straight lines along the row or column they are placed at, and can't jump over other pieces. For a more detailed explanation of the moves of rooks, along with images, please [click here](#).

Input Format

- The first line contains TT - the number of test cases. Then the test cases follow.
- The first line of each test case contain four space-separated integers each X_1, Y_1, X_2, Y_2 - $(X_1,Y_1)(X_1,Y_1)$ is the position of the first rook and $(X_2,Y_2)(X_2,Y_2)$ is the position of the second rook.

Output Format

For each test case, output on a single line YES (without quotes) if the rooks attack each other, and NO otherwise.

You may print each character of the string in uppercase or lowercase (for example, the strings Yes, YEs, yes and yeS will all be treated as identical).

Solution :

```
t = int(input())
for _ in range(t):
    m = list(map(int,input().split()))[:4] # moves
    if( m[0]==m[2] or m[1]==m[3] ):
        print("YES")
    else:
        print("NO")
```

Test Cases:**Input:**

```

3
2 3 4 5
3 3 3 3
2 3 1 2

```

Output :

```

YES
YES
NO

```

Result :

The screenshot displays the CodeChef interface for the 'Two Rooks' problem. The problem statement describes an 8x8 chessboard with 2 rooks and asks if they attack each other based on their coordinates. A Python solution is provided, which reads the number of test cases, then for each test case, reads four integers representing the positions of two rooks. It checks if they share the same row or column and prints 'YES' or 'NO' accordingly. The submission result shows a 'Correct Answer' status with a submission ID of 73061326 and a time of 0.05s.

Problem Statement:

You are given a standard 8×8 chessboard which has exactly 2 rooks on it and no other pieces. The rows are numbered 1 to 8 from bottom to top, and the columns are numbered 1 to 8 from left to right. The cell at the intersection of the i -th column and j -th row is denoted (i, j) .

Given the initial positions of the rooks in the form of coordinates (X_1, Y_1) and (X_2, Y_2) , you need to tell whether the 2 rooks currently attack each other or not. Assume, each square can contain at most one piece.

Rooks can only travel in straight lines along the row or column they are placed at, and can't jump over other pieces. For a more detailed explanation of the moves of rooks, along with images, please [click here](#).

Input Format

- The first line contains T - the number of test cases. Then the test cases follow.
- The first line of each test case contain four space-separated integers each X_1, Y_1, X_2, Y_2 - (X_1, Y_1) is the position of the first rook and (X_2, Y_2) is the position of the second rook.

Python Solution:

```

1 t = int(input())
2
3 for _ in range(t):
4     m = list(map(int, input().split()))[:4] # moves
5
6     if (m[0]==m[2] or m[1]==m[3]):
7         print("YES")
8     else:
9         print("NO")
10

```

Submission Result:

Status: ✔ Correct Answer Submission ID: 73061326

Time: 0.05s

Congratulations on solving the problem. Visit our [practice section](#) to solve more interesting problems

Why only this DS is used? :

This question is all about conditional statements (if, else) successfully executed.

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

WEEK – 3**Problem**

Suppose Chef is stuck on an island and currently he has xx units of food supply and yy units of water supply in total that he could collect from the island. He needs x_rxr units of food supply and y_ryr units of water supply per day at the minimal to have sufficient energy to build a boat from the woods and also to live for another day. Assuming it takes exactly DD days to build the boat and reach the shore, tell whether Chef has the sufficient amount of supplies to be able to reach the shore by building the boat?

Input:

- First line will contain TT , number of testcases. Then the testcases follow.
- Each testcase contains of a single line of input, five integers x, y, x_r, y_r, D .

Output:

For each testcase, output in a single line answer "YES" if Chef can reach the shore by building the boat and "NO" if not (without quotes).

You may print each character of each string in uppercase or lowercase (for example, the strings "yEs", "yes", "Yes" and "YES" will all be treated as identical).

Solution :

```
for _ in range(int(input())):
    x,y,x1,y1,d = map(int,input().split())
    food = x/x1
    water = y/y1
    mindays = min(food,water)
    # print(mindays)
    # print(d)
    if d<=mindays:
        print("YES")
    else:
        print("NO")
```

Test Cases:**Input:**

```
3
4 2 1 1 1
4 2 1 3 1
4 2 4 2 2
```

Output :

```
YES
NO
NO
```

Result :

The screenshot shows the CodeChef interface for a problem. The problem statement describes a scenario where Chef needs to build a boat to reach the shore, given constraints on food and water supply and the time to build the boat. The input consists of the number of test cases followed by five integers per test case. The output is a single line answer "YES" or "NO" for each test case. The submission result shows a "Correct Answer" status with a submission ID of 73848320, a time of 0.02s, and a subtask score of 100.00%.

Statement Hints Submissions Solution Ask a Doubt

Input:

- First line will contain T , number of testcases. Then the testcases follow.
- Each testcase contains of a single line of input, five integers x, y, x_r, y_r, D .

Output:

For each testcase, output in a single line answer "YES" if Chef can reach the shore by building the boat and "NO" if not (without quotes).

You may print each character of each string in uppercase or lowercase (for example, the strings "yEs", "yes", "Yes" and "YES" will all be treated as identical).

Status: ✓ Correct Answer Submission ID: [73848320](#)

Time: 0.02s

Sub-Task	Task #	Result (time)
1	0	AC (0.023461)
Subtask Score: 100.00%		Result - AC
Total Score = 100.00%		

Why only this DS is used? :

This question is all about conditional statements (if, else) successfully executed.

Time Complexity : $O(n)$

Space Complexity : $O(n)$

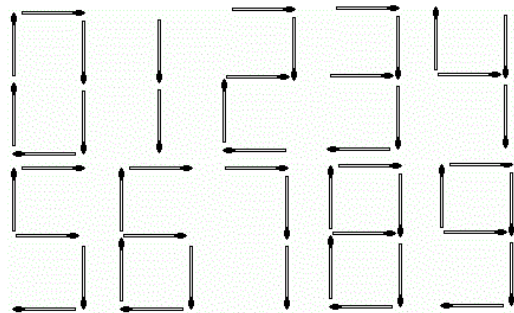
Result : Successfully Executed the Program.

WEEK – 4**Task 1****Problem**

Chef's son Chefu found some matches in the kitchen and he immediately starting playing with them.

The first thing Chefu wanted to do was to calculate the result of his homework — the sum of AA and BB , and write it using matches. Help Chefu and tell him the number of matches needed to write the result.

Digits are formed using matches in the following

**Input**

- The first line of the input contains a single integer TT denoting the number of test cases. The description of TT test cases follows.
- The first and only line of each test case contains two space-separated integers AA and BB .

Output

For each test case, print a single line containing one integer — the number of matches needed to write the result $(A+B)$.

Solution:

cook your dish here

```
nsticks = [6,2,5,5,4,5,6,3,7,6]
```

```
for _ in range(int(input())):
```

```
    n = 0
```

```
    a,b = map(int,input().split())
```

```
    c = a+b
```

```
    c = str(c)
```

```
    # print(c)
```

```
    for i in c:
```

```
# print("val[{}] : {}".format(int(i),nsticks[int(i))))

n = n+nsticks[int(i)]

print(n)
```

Test Cases

Input:

```
3
123 234
10101 1010
4 4
```

Output :

```
13
10
7
```

Result:

The screenshot displays the CodeChef interface for the 'Playing with Matches' problem. The problem statement asks for the number of matches needed to write the result of $A + B$ for each test case. The constraints are $1 \leq T \leq 1,000$ and $1 \leq A, B \leq 10^6$. The sample input and output are as follows:

Input	Output
3	13
123 234	10
10101 1010	7
4 4	

The submission results show a 'Correct Answer' status with a submission ID of 76558555. The time taken for the submission was 0.03s. The problem solver badge indicates that 48 out of 50 problems have been solved, and the user is 2 problems away from getting a Bronze Badge.

Why only this DS is used?

Counting no of sticks used to build that number using a list.

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

Task 2 :**Problem**

You are given a binary string AA of length NN .

You can perform the following type of operation on the string AA :

- Choose two **different** indices ii and jj ($1 \leq i, j \leq N$);
- Change $A_i A_i$ and $A_j A_j$ to $A_i \oplus A_j A_i \oplus A_j$. Here \oplus represents the [bitwise XOR](#) operation.

Find the **minimum** number of operations required to convert the given string into a [palindrome](#).

Input Format

- First line of the input contains TT , the number of test cases. Then the test cases follow.
- First line of each test case contains an integer NN denoting the size of the string.
- Second line of each test case contains a binary string AA of length NN containing 00s and 11s only.

Output Format

For each test case, print the **minimum** number of operations required to make the string a palindrome.

Solution

```
for _ in range(int(input())):
    n = int(input())
    a = input()
    if n==1:
        count = 0
    else:
        count = 0
        for i in range(n//2):
            if a[i]!=a[n-i-1]:
                #xor = int(a[i]) ^ int(a[-i])
                #a[i] = str(xor)
                #a[-i] = str(xor)
                #print("doesn't match",a[i],a[-i])
                count = count+1
```

```
print((count+1)//2)
```

Test Cases

Input:

4
5
11011
7
0111010
1
1

4
1100

Output :

0
1
0
1

Result

The screenshot shows the CodeChef submission page for the 'Xor Palindrome' problem. The problem statement is visible, along with sample input and output. The submission status is 'Correct Answer' with a submission ID of 76559549. The execution time is 0.04s. The subtask results show a score of 100.00% for all 6 subtasks.

Sub-Task	Task #	Result (time)
1	2	AC (0.041442)
1	3	AC (0.036156)
1	4	AC (0.036017)
1	5	AC (0.041190)
1	6	AC (0.037172)

Subtask Score: 100.00% Result - AC

Why only this DS is used?

No special data structure is not used here only a simple logic which is counting no of elements which spoil palindrome and divide it by 2 by adding 1 to sum

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

Task 3 :**Problem**

Chef has N small boxes arranged on a line from 1 to N . For each valid i , the weight of the i -th box is W_i . Chef wants to bring them to his home, which is at the position 0. He can hold any number of boxes at the same time; however, the total weight of the boxes he's holding must not exceed K at any time, and he can only pick the i -th box if all the boxes between Chef's home and the i -th box have been either moved or picked up in this trip.

Therefore, Chef will pick up boxes and carry them home in one or more round trips. Find the smallest number of round trips he needs or determine that he cannot bring all boxes home.

Input

- 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 first line of each test case contains two space-separated integers N and K .
- The second line contains N space-separated integers W_1, W_2, \dots, W_N .

Output

For each test case, print a single line containing one integer — the smallest number of round trips or -1 if it is impossible for Chef to bring all boxes home.

Solution**# cook your dish here**

```
for _ in range(int(input())):
    n,k=map(int,input().split())
    l=list(map(int,input().split()))[:n]
    if max(l)>k:
        print("-1")
    else:
        count = 0
        s = 0
        for i in range(n) :
            if s+l[i]<=k:
                s+=l[i]
            else:
```

```

s=l[i]

count+=1

print(count+1)

```

Test Cases

Input:

```

4
1 1
2
2 4
1 1
3 6
3 4 2

```

```

3 6
3 4 3

```

Output :

```

-1
1
2
3

```

Result

codechef.com/submit/CHEFNWRK?tab=statement

Chef and Work Difficulty Rating: 1185 Expand

Prev Problem Next Problem

Statement Hints Submissions Solution Ask a Doubt

number of round trips or -1 if it is impossible for Chef to bring all boxes home.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N, K \leq 10^3$
- $1 \leq W_i \leq 10^3$ for each valid i

Sample 1:

Input	Output
4	-1
1 1	1
2	2
2 4	3
1 1	
3 6	
3 4 2	
3 6	
3 4 3	

PYTH 3

Upload code as file Compile & Run Submit Code

Problem Solver Badge 49 / 50

Solve 1 more problem to get Bronze Badge

Status: ✓ Correct Answer Submission ID: 76579924

Time: 0.05s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Why only this DS is used?

No data structure as such we are just using a list to input values and applying logic on it with a for loop

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

WEEK – 5**Task 1****Problem**

On a sunny day, Akbar and Birbal were taking a leisurely walk in palace gardens. Suddenly, Akbar noticed a bunch of sticks on the ground and decided to test Birbal's wits.

There are NN stick holders with negligible size (numbered 11 through NN) in a row on the ground. Akbar places all the sticks in them vertically; for each valid ii , the initial height of the stick in the ii -th holder is A_iA_i . Birbal has a stick cutter and his task is to completely cut all these sticks, i.e. reduce the heights of all sticks to 00. He may perform zero or more operations; in each operation, he should do the following:

- Choose an integer HH and fix the cutter at the height HH above the ground.
- The cutter moves from the 11-st to the NN -th stick holder. Whenever it encounters a stick whose current height is greater than HH , it cuts this stick down to height HH (i.e. for a stick with height $h \setminus gt Hh>H$, it removes its upper part with length $h-Hh-H$).
- All the upper parts of sticks that are cut in one operation must have equal lengths. Otherwise, the operation may not be performed.

For example, if the heights of sticks are initially $[5, 3, 5][5,3,5]$, then some valid values for HH in the first operation are 33 and 44 — the cutter cuts the upper parts of two sticks and their lengths are $[2, 2][2,2]$ and $[1, 1][1,1]$ respectively. $H = 2H=2$ is an invalid choice because it would cut the upper parts of all three sticks with lengths $[3, 1, 3][3,1,3]$, which are not all equal.

Akbar wants Birbal to completely cut all sticks in the minimum possible number of operations. If you want to be friends with Birbal, help him solve the problem.

Input

- The first line of the input contains a single integer TT denoting the number of test cases. The description of TT test cases follows.
- The first line of each test case contains a single integer NN .
- The second line contains NN space-separated integers $A_1, A_2, \ldots, A_NA1,A2,...,AN$.

Output

For each test case, print a single line containing one integer — the minimum number of operations needed to completely cut all the sticks.

Solution:

```
t = int(input())

for i in range(t):

    n = int(input())

    l = list(map(int, input().split()))
```

```
l = list(set(l))
```

```
x = l.count(0)
```

```
print(len(l) - x)
```

Test Cases

Input:

1

3

1 2 3

Output:

3

Result:

The screenshot shows the CodeChef submission interface for the problem "A Problem on Sticks". The problem statement describes a task where sticks of different heights are placed in holders, and the goal is to cut them to a common height H using a stick cutter. The submission result shows that the code was executed successfully, with a status of "Correct Answer" and a submission ID of 79365222. The execution time was 0.44s. The result table shows two sub-tasks, both of which were completed successfully (AC) with scores of 20.00% and 80.00% respectively, resulting in a total score of 100.00%.

Sub-Task	Task #	Result (time)
1	1	AC (0.023394)
Subtask Score: 20.00%		Result - AC
2	2	AC (0.436886)
Subtask Score: 80.00%		Result - AC
Total Score = 100.00%		

Why only this DS is used?

No special data structure is not used here only a simple logic by using a for loop

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

Task 2**Problem**

Chef is very hungry. So, Chef goes to a shop selling burgers. The shop has 22 types of burgers:

Normal burgers, which cost XX rupees each

Premium burgers, which cost YY rupees each (where $Y > X$)

Chef has RR rupees. Chef wants to buy exactly NN burgers. He also wants to maximize the number of premium burgers he buys. Determine the number of burgers of both types Chef must buy.

Output -1-1 if it is not possible for Chef to buy NN burgers.

Input Format

The first line contains a single integer TT — the number of test cases. Then the test cases follow.

The first and only line of each test case contains four space-separated integers XX, YY, NN and RR — the cost of a normal burger, the cost of a premium burger, the number of burgers Chef wants to buy and the amount of money Chef has.

Output Format

For each test case, output on a new line two integers: the number of normal burgers and the number of premium burgers Chef must buy satisfying the given conditions.

Output -1-1 if he cannot buy NN burgers.

Solution

```
t = int(input())
for _ in range(t):
    x,y,n,r = map(int,input().split())
    if y*n <= r:
        print(0,n)
    elif x*n > r:
        print(-1)
    else:
        z = (r - (n*x)) // (y-x)
```

```
print((n - z), z)
```

Test Cases

Input:

4
2 10 4 12
4 8 10 50
99 100 5 10
9 10 10 200

Output:

4 0
8 2
-1
0 10

Result :

The screenshot shows a web browser window with the URL `codechef.com/submit/BURGERS2`. The page title is "Hungry Chef" with a difficulty rating of 1187. The submission status is "Correct Answer" with a submission ID of 79366168. The execution time is 0.08s. A table shows the results for 5 sub-tasks, all of which are "AC" (Accepted).

Sub-Task	Task #	Result (time)
1	0	AC (0.022254)
1	1	AC (0.070957)
1	2	AC (0.075365)
1	3	AC (0.075719)
1	4	AC (0.077346)

Subtask Score: 100.00% Result - AC

Why only this DS is used?

No special data structure is not used here only a simple logic by using a for loop and some conditional statements.

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.

Task 3**Problem**

Chef is given a binary string AA of length NN . He can perform the following operation on AA any number of times:

- Choose LL and RR ($1 \leq L \leq R \leq N$), such that, in the [substring](#) $A[L,R]A[L,R]$, the number of 1s is **equal** to the number of 00s and **reverse** the substring $A[L,R]A[L,R]$.

Find the lexicographically **smallest** string that Chef can obtain after performing the above operation **any** (possibly zero) number of times on AA .

String XX is lexicographically smaller than string YY , if either of the following satisfies:

- XX is a prefix of YY and $X \neq Y$.
- There exists an index i such that $X_i < Y_i$ and $X_j = Y_j$, $\forall j$ such that $1 \leq j < i$.

Input Format

- First line will contain T , the number of test cases. Then the test cases follow. Each test case contains two lines.
- The first line contains the integer N , the length of the binary string.
- The second line contains the binary string AA .

Output Format

For each test case, print the lexicographically **smallest** binary string that can be obtained after performing the operation **any** (possibly zero) number of times.

Solution:

```
n=int(input ())
for i in range(n):
    t=int(input ())
    a=input ()
    x=a.count('0')
    y=a.count('1')
    print (x*'0'+y*'1')
```

Test Cases

Input:

2
5
01100
4
0000

Output:

00011
0000

Result:

codechef.com/submit/BALREV

Balanced Reversals Difficulty Rating: 1165 Expand

Prev Problem Next Problem

Statement Hints Submissions Solution Ask a Doubt PYTH 3

The second line contains the binary string A .

Output Format
For each test case, print the lexicographically **smallest** binary string that can be obtained after performing the operation **any** (possibly zero) number of times.

Constraints

- $1 \leq T \leq 100$
- $1 \leq N \leq 10^5$
- Sum of N over all test cases does not exceed $2 \cdot 10^5$.

Sample 1:

Input	Output
2	00011
5	0000
01100	

Status: ✓ Correct Answer Submission ID: [79366924](#)

Time: 0.03s

Sub-Task	Task #	Result (time)
1	1	AC (0.024030)
1	2	AC (0.025195)
1	3	AC (0.022833)

Subtask Score: 100.00% Result - AC

Upload code as file Compile & Run Submit Code

Why only this DS is used?

No special data structure is not used here only a simple logic by using a for loop and some conditional statements and mathematical expressions.

Time Complexity : $O(n)$

Space Complexity : $O(n)$

Result : Successfully Executed the Program.