

שאלות בנוגע לתרגיל נא להפנות דרך פורום הקורס שנפתח במיוחד לשם כך:

<https://piazza.com/biu.ac.il/fall2018/89110>

אם לא נענתה תשובה תוך 24 שעות, נא לשלוח אלי (דור) מייל עם לינק לדיון הרלוונטי ואענה. המייל הוא: [nisimdor@gmail.com](mailto:nisimdor@gmail.com). בכל מייל יש לציין שם, שם משתמש, מס' קורס, וקבוצת תרגול.

- מועד פרסום: 18/11/2018
- מועד אחרון להגשה: 25/11/2018
- יש לשלוח את הקבצים באמצעות האתר: <http://help.cs.biu.ac.il/submit.htm>
- שם ההגשה של התרגיל: ex4
- יש להקפיד מאוד על כל הוראות עיצוב הקלט והפלט, כמפורט בכל סעיף וסעיף. על הפלט להיראות בדיוק כמו בדוגמאות. אין להוסיף או להשמיט רווחים או תווים אחרים ואין להחליף אותיות גדולות בקטנות או להיפך ☹ אי-הקפדה על פרטים אלה עלולה לגרור ירידה משמעותית ביותר בציון התרגיל עד כדי 0. ראו הוזהרתם!
- להזכירכם, העבודה היא אישית. "עבודה משותפת" דינה כהעתקה.
- אין להדפיס למסך שום דבר מעבר למה שנתבקש בתרגיל.
- יש לוודא שהתרגיל מתקמפל ורץ על השרתים באוניברסיטה (u2) ללא שגיאות/אזהרות.
- אתם יכולים לעבוד עם כל עורך טקסטואלי שאתם מעדיפים. להזכירכם – pico בשרתי linux שבמעבדות; notepad ב-windows; או בסביבת פיתוח ייעודית.

**הקפידו על כתיבה לפי קובץ ה-Coding-Style שבאתר הקורס!!**

## הוראות כלליות לתרגיל

כתבו תכנית בעלת הזרימה הבאה:

1. הדפס: Please enter the mission number:
2. קלוט מספר שלם (ניתן להניח שיתקבל מספר שלם)
3. כל עוד המספר בין 1 ל-4 בצע:
  - i. אם התקבל 1 בצע את משימה 1.
  - ii. אם התקבל 2 בצע את משימה 2.
  - iii. אם התקבל 3 בצע את משימה 3.
  - iv. אם התקבל 4 בצע את משימה 4.
- b. הדפס: Please enter the mission number:
- c. קלוט מספר שלם (ניתן להניח שיתקבל מספר שלם)
4. צא מהתוכנית

ניתן להניח שכל קלט המתקבל במשימות הינו **תקין** אלא אם כן נאמר אחרת!  
**יש להשתמש בלבד בחומר הנלמד עד כה בתרגולים (עד פונקציות ורקורסיות כולל).**

## אסור להשתמש במשתנים גלובליים וסטטיים!!!

- כאשר מבקשים שפונקציה תהיה רקורסיבית, המשמעות היא **אסור לולאות**
- מאחר ולמדנו על פונקציות, נסו לחשוב על פתרון למטלה תוך שימוש בפונקציות.
- תחשבו אולי להגדיר את ה-Main באופן הבא:

```
void main()
{
    int mission;
    do{
        scanf("%d", &mission);
        switch(mission){
            case 1: Mission1();
                break;
            .....
        }
    }while(mission >= 1 && mission <= 4);
}
```

## שים/י ♥

כאשר מתרחשת שגיאה בקוד (משמע, הודפסה הודעת שגיאה) עליכם לחזור לתוכנית הראשית **ולא** לבצע את המשימה מחדש!

You are given  $N$  Queries. Each query consists of two numbers  $a$  and  $b$ .

Determine the number of pairs that satisfy the following condition:

The number  $a$  contains the number  $b$ .

You can assume that  $a \geq b$

### **Input Format**

The input format will be as follows:

Please enter the number of pairs:

<N>

Please enter the larger number of pair 1 (a):

<a<sub>1</sub>>

Please enter the smaller number of pair 1 (b):

<b<sub>1</sub>>

.....

Please enter the larger number of pair <N> (a):

<a<sub>N</sub>>

Please enter the smaller number of pair <N> (b):

<b<sub>N</sub>>

The first line contains the integer  $N$ .

The next  $2 \times N$  lines each contains the pairs of integers,  $a$  &  $b$ .

### **Constraints**

$$1 \leq N \leq 10^3$$

$$1 \leq a, b \leq 10^9$$

### **Output Format**

The output format will be as follows:

The number of pairs satisfying the condition is: <result><newline>

- <result> - the number of pairs satisfying the condition.

### **Sample Input**

Please enter the number of pairs:

3

Please enter the larger number of pair 1 (a):

171188118

Please enter the smaller number of pair 1 (b):

**881**

Please enter the larger number of pair 2 (a):

**34221**

Please enter the smaller number of pair 2 (b):

**21**

Please enter the larger number of pair 3 (a):

**123456**

Please enter the smaller number of pair 3 (b):

**46**

### Sample Output

The number of pairs satisfying the condition is: **2**<newline>

### Explanation

For test case 1, the number 1711**881**18 contains the number **881** and thus the pair is counted.

For the case 2, the number 342**21** contains the number **21** and thus the pair is counted

For the case 3, the number 123456 does not contain the number **46** and thus the pair is not counted.

Thus the output is **2**.

## All the functions in this mission must be implemented using a Recursion!

You are given a single number  $N$ . You can perform any of the two operations on  $N$  in each move:

1: If we take 2 integers  $a$  and  $b$  where  $N = a \times b (a \neq 1, b \neq 1)$  then we can change  $N = \max(a, b)$

2: Decrease the value of  $N$  by 1.

In the event where either action produces the same number of future moves **Action number 1** is preferable.

Determine the minimum number of moves required to reduce the value of  $N$  to 0 .

### Input Format

The input format will be as follows:

Please enter the number:

<N>

- <N> - The integer N that is examined.

### Constraints

$$0 \leq N \leq 300$$

### Output Format

The input format will be as follows:

The sequence of numbers produced by the actions is:

<n<sub>1</sub>>

....

<n<sub>k-1</sub>>

0

The minimum number of moves is <K><newline>

- <n<sub>i</sub>> - The number created after performing action **i**.
- <K> - The minimum number of moves required to reduce the value of N to 0.

### Sample Input

Please enter the number:

**3**

### Sample Output

The sequence of numbers produced by the actions is:

2

1

0

The minimum number of moves is **3**<newline>

### Explanation

We only have one option that gives the minimum number of moves.

Follow  $3 \rightarrow 2 \rightarrow 1 \rightarrow 0$ . Hence, 3 moves.

### Sample Input

Please enter the number:

**150**

### Sample Output

The sequence of numbers produced by the actions is:

75

25

5

4

2

1

0

The minimum number of moves is **7**<newline>

### Explanation

We only have one option that gives the minimum number of moves.

Follow  $75 \rightarrow 25 \rightarrow 5 \rightarrow 4 \rightarrow 2 \rightarrow 1 \rightarrow 0$ . Hence, 7 moves.

## All the functions in this mission must be implemented using a Recursion!

Given a long unsigned integer  $x$ , count the number of values of  $a$  satisfying the following conditions:

- $a \oplus x > x$
- $0 < a < x$

where  $a$  and  $x$  are long integers and  $\oplus$  is the [bitwise XOR](#) operator.

You are given a long integer  $x$ .

Print all the values of  $a$  (**in an ascending order**) satisfying the conditions above on a new line.

For example, you are given the value  $x = 5$ . Condition 2 requires that  $0 < a < x$ . The following tests are run:

- 1  $\oplus 5 = 4$
- 2  $\oplus 5 = 7$
- 3  $\oplus 5 = 6$
- 4  $\oplus 5 = 1$

We find that there are 2 values meeting the first condition: 2 and 3.

### Input Format

The input format will be as follows:

Please enter the number:

<x>

- <x> - the value of  $x$  (long unsigned integer)

### Constraints

- $1 \leq x \leq 10^2$

### Output Format

The output format will be as follows:

The numbers are:

<a<sub>1</sub>>

...

<a<sub>k</sub>>

A total of <K> numbers <newline>

<a<sub>i</sub>> - The  $i$  values found in an **Ascending order** that satisfies the conditions

<K> - The number of values of **a** satisfying the given conditions

If there aren't any numbers the output will be:

The numbers are:

A total of **0** numbers <newline>

### Sample Input 0

Please enter the number:

**2**

### Sample Output 0

The numbers are:

**1**

A total of **1** numbers <newline>

### Explanation 0

For  $x = 2$  the only value of **a** satisfying  $0 < a < x$  is 1 . This also satisfies our other condition, as  $1 \oplus 2 = 3$  and  $3 > x$ . Because we have one valid **a** and there are no more values to check, we print **1** on a new line.

### Sample Input 1

Please enter the number:

**10**

The numbers are:

**1**

**4**

**5**

**6**

**7**

A total of **5** numbers <newline>

### Explanation 0

For  $x=10$  , the following values of **a** satisfy our conditions:

$$1 \oplus 10 = 11$$

$$4 \oplus 10 = 14$$

$$5 \oplus 10 = 15$$

$$6 \oplus 10 = 12$$

$$7 \oplus 10 = 13$$

There are five valid values of a.



## Rock, Paper, Scissors - 4 משימה

Rock, Paper, Scissors, Lizard, Spock is a game, invented by Sam Kass and Karen Bryla, that extends the Rock, Paper, Scissors game. It is a favorite of the character Sheldon from the TV Show, *The Big Bang Theory*. [Video Link](#)

The rules of the game are as follows. Each player chooses a shape from the set {*Rock, Paper, Scissors, Lizard, Spock*}. If the two players choose the same shape, they tie. Otherwise, the winner is the one with the shape listed first in the following set of rules:

Scissors cuts Paper  
Paper covers Rock  
Rock crushes Lizard  
Lizard poisons Spock  
Spock smashes Scissors  
Scissors decapitates Lizard  
Lizard eats Paper  
Paper disproves Spock  
Spock vaporizes Rock  
(and as it always has) Rock crushes scissors

Below is a graphical representation from *The Big Bang Theory's* wiki:



Alice and Bob also enjoy this game, and they have decided to play a series of them. Each has a very specific strategy that they follow.

Alice's strategy is as follows:

- 1) If she wins a game, she keeps the same shape.
- 2) If she ties, she chooses a shape from one of the two that would beat her current shape. Of these two, she chooses the one that beats the other. For example, if she has tied when choosing *Rock*, her options are *Paper* and *Spock*. Since *Paper* beats *Spock*, she chooses *Paper*.
- 3) If she loses, she chooses a shape from one of the two that would beat her opponent's current shape. Of these two, she chooses the one that beats the other. For example, let's say she has lost by choosing *Rock*, when her opponent chose *Paper*.

She will then choose from *Scissors* or *Lizard*. Since *Scissors* beats *Lizard*, she chooses *Scissors*.

Bob's strategy is as follows:

- 1) Every other turn, he chooses *Spock*.
- 2) If he won the previous turn when playing *Spock*, he chooses *Rock*.
- 3) If he tied the previous turn when playing *Spock*, he chooses *Lizard*.
- 4) If he lost the previous turn when playing *Spock*, he chooses *Paper*.

Your task is to write a program that evaluates a series of games between Alice and Bob.

### Input Format

The input format will be as follows:

Please enter the shape Alice chose:

**<AliceShape>**

Please enter the shape Bob chose:

**<BobShape>**

Please enter the number of rounds:

**<n>**

- **<AliceShape>** and **<BobShape>** are values from the given ENUM:

```
enum { Rock = 0, Paper = 1, Scissors = 2, Lizard = 3, Spock = 4}
```

- **<AliceShape>** is the shape that Alice will choose in the first game of the series.
- **<BobShape>** is the shape that Bob will choose in the first game of the series.
- **<n>** - An integer indicating how many games Alice and Bob will play in the series.

### Constraints

$$1 \leq n \leq 30$$

- You must create a `#define` for the strings of all the shapes in addition to the definitions of the max and minimum **n** values!
  - **Rock**
  - **Paper**
  - **Scissors**

- **Lizard**
- **Spock**

### Output Format

The output format will be as follows:

```
<AliceShapei> <opi> <BobShapei>
.....
<AliceShapen> <opn> <BobShapen>
```

- <AliceShape<sub>i</sub>> - The **string** of the shape that Alice played in round **i**.
- <BobShape<sub>i</sub>> - The **string** of the shape that Bob played in round **i**.
- <op<sub>i</sub>> - Will a character out of the following options:
  - **<** if Bob won in round **i**.
  - **>** if Alice won in round **i**.
  - **=** if there was a tie between Bob and Alice.

Then one of the following lines:

**[Player] wins, by winning [WinGames] game(s) and tying [TieGames] game(s)**

**Or:** (if Bob and Alice had a tie)

**Alice and Bob tie, each winning [WinGames] game(s) and tying [TieGames] game(s)**

- [Player] is the name of the player with more wins (either "Alice" or "Bob")
- [WinGames] is the number of games won either by the winner or, in the case of a tie, by each player
- [TieGames] is the number of games in which the players tied

Notes:

- The output is case sensitive. The player names, for example, must be either "Alice" or "Bob". Neither "alice" nor "BOB" will be acceptable.
- The words are separated by a single space, and there are no spaces before the first word in the line, nor after the last word in the line.

### Sample Input 0

Please enter the shape Alice chose:

**0**

Please enter the shape Bob chose:

**4**

Please enter the number of rounds:

4

### Sample Output

Rock < Spock

Lizard < Rock

Paper > Spock

Paper = Paper

Bob wins, by winning 2 game(s) and tying 1 game(s)

### Explanation

In the first game, Bob wins, since *Spock* vaporizes *Rock*.

*Bob* won when choosing *Spock* so he chooses *Rock*. *Alice* lost, so she chooses from *Paper* and *Lizard*, both of which beat *Bob's* last choice of *Spock*.

Since *Lizard* beats *Paper*, she chooses *Lizard*.

In the second game, then, Bob wins again, because *Rock* crushes *Lizard*.

*Bob* did not play *Spock* last turn, so he chooses *Spock* next. *Alice* lost, so she chooses from *Paper* and *Spock*, both of which beat *Bob's* last choice of *Rock*.

Since *Paper* beats *Spock*, she chooses *Paper*.

In the third game, *Alice* wins, since *Paper* disproves *Spock*.

*Bob* lost when choosing *Spock* so he chooses *Paper*. *Alice* won, so she continues playing *Paper*.

In the fourth game, they tie by both choosing *Paper*.

### Sample Input 1

Please enter the shape Alice chose:

1

Please enter the shape Bob chose:

1

Please enter the number of rounds:

1

### Sample Output

Paper = Paper

Alice and Bob tie, each winning 0 game(s) and tying 1 game(s)

### Explanation

This test case consists of a single game in which both players play *Paper*.

Please enter the mission number:

**1**

Please enter the number of pairs:

3

Please enter the larger number of pair 1 (a):

171188118

Please enter the smaller number of pair 1 (b):

881

Please enter the larger number of pair 2 (a):

34221

Please enter the smaller number of pair 2 (b):

21

Please enter the larger number of pair 3 (a):

123456

Please enter the smaller number of pair 3 (b):

46

The number of pairs satisfying the condition is: 2

Please enter the mission number:

**2**

Please enter the number:

3

The sequence of numbers produced by the actions is:

2

1

0

The minimum number of moves is 3

Please enter the mission number:

**2**

Please enter the number:

150

The sequence of numbers produced by the actions is:

75

25

5

4

2

1

0

The minimum number of moves is 7

Please enter the mission number:

3

Please enter the number:

2

The numbers are:

1

A total of 1 numbers

Please enter the mission number:

3

Please enter the number:

10

The numbers are:

1

4

5

6

7

A total of 5 numbers

Please enter the mission number:

4

Please enter the shape Alice chose:

0

Please enter the shape Bob chose:

4

Please enter the number of rounds:

4

Rock < Spock

Lizard < Rock

Paper > Spock

Paper = Paper

Bob wins, by winning 2 game(s) and tying 1 game(s)

Please enter the mission number:

4

Please enter the shape Alice chose:

1

Please enter the shape Bob chose:

1

Please enter the number of rounds:

1

Paper = Paper

Alice and Bob tie, each winning 0 game(s) and tying 1 game(s)

Please enter the mission number:

5