CCSCNE-2026: Programming Contest

April 10^{th}, 2026

1. Iterated Shuffle

The *shuffle* operation that can be applied to lists of length n, which, for each i from 0 to n repeats the following:

1. If $\lfloor i \cdot 2.5 \rfloor \geq n - i - 1$, do nothing.
2. Otherwise, swap the elements at positions $\lfloor i \cdot 2.5 \rfloor$ and $n - i - 1$.

Your task is to write a program that finds the smallest, non-trivial answer to the question, "Given an arbitrary list, how many times must the *shuffle* operation be iterated until the list is back to its original order?" (Note that a trivial answer is 0 for any list, but since it is an uninteresting answer, we will ignore it.) For example, if the list is $[0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]$, 12 *shuffle* iterations are required to put the list back to the original order.

Input

1. Each list will contain only integers in the range $[0, 999]$.
2. Each list will be on a separate line
3. There will be a single space between each integer in every list.
4. The end of input will be signaled with a -1 on a line by itself.

```
0 1 2 3 4 5
0 1 2 3 4 5 6
0 1 2 3 4 5 6 7 8 9
0 0 0 1 1 1 0 0 0 0
-1
```

Output

1. For each test case output the number of iterations of *shuffle* required to put the list back in its original order.

```
6
3
12
4
```

2. Non-Overlapping Triangles

A *triangle* of integers is a collection of three integers where a sum of any two integers in the collection is greater than the third. For example, the collection 2, 2, 3 is a triangle, because $2+2>3$ and $2+3>2$, but the collection 2, 2, 4 is not a triangle, because $2+2 \not> 4$.

Two triangles *overlap* if one of the integers in one triangle falls between some of the numbers from the other triangle. For example, the triangles 2,4,5 and 3,6,8 overlap because the 3 from the second triangle is between the 2 and the 5 from the first triangle. Triangles that do not overlap are called *non-overlapping*; an example of two such triangles is 2,4,5 and 6,12,17, because none of the integers 2,4,5 are between 6 and 17 and none of the integers 6,12,17 are between 2 and 5.

Your task is to write a program that accepts several lists of integers as input and prints the maximum number of non-overlapping triangles that can be made from each list.

Input

1. Each list will contain only integers in the range $[1, 1000000]$.
2. Each list will be on a separate line
3. There will be a single space between each integer in every list.
4. The end of input will be signaled with a -1 on a line by itself.

```
32 3 33 34 2 31
36 35 34 4 33 32 2 31 3
2 2 4 31 32 33 34 35 36
-1
```

Output

1. For each list output the maximum number of non-overlapping triangles that can be formed using the numbers from the list.

```
1
3
2
```

3. Kaprekar's Routine

Kaprekar's routine is an algorithm where:

1. Given a four digit number X
2. Derive two numbers X_1 and X_2 from it by sorting the digits of X in ascending and descending order, respectively.
3. Subtract: $Y = X_1 - X_2$
4. Quit if Y is equal to 0 or 6174, otherwise set X = Y and repeat

Your program will answer the question of how many loops does it take to reach either 0 or 6174 for a list of numbers.

Input

1. The numbers will each be on a separate line.
2. Numbers with less than four digits should be processed as if they had enough leading 0s to reach four digits.
3. The end of input will be signaled with a -1 on a line by itself.

```
3172
5116
8982
3334
2187
4444
-1
```

Output

1. For each number **only** print the number of times through the loop on a separate line.
2. At the end of input print the word **done** on a separate line.

```
7
5
4
5
5
1
done
```

4. Shape Recognition

Given a 24x24 grid of ascii characters, a **shape** is defined as a set of cells that:

1. All contain the same character.
2. Are adjacent to each other in the four cardinal directions.
3. Are maximal, i.e., it is not part of a larger shape.

Your program should find the number of **unique** shapes of a given size.

Please Note:

1. Two shapes are considered identical if one can be translated (shifted horizontally and/or vertically) to perfectly overlap the other.
2. Two shapes are identical if one of them can be rotated by 90, 180, or 270 degrees to perfectly overlap with the other.
3. Two shapes can be identical even if they consist of different characters.

Input

1. The number of test cases will be on a line by itself.
2. Each test case:
   (a) 24 lines of ascii characters with shape data
   (b) 1 line with the target shape size

```
2
AAAAAAAAAAAAAAAAAAAAAAAA
A.BBBB..CCCC..DDDD..EEEE
A.B.......C.....D......E
A.B..FF...C..GG.D..HH..E
A.B..FF...C..GG.D..HH..E
A.......................E
A..IIII..JJJJ..KKKK....E
A.......................E
A..LL....MM......NN....E
A..LL....MM......NN....E
A..LL....MM......NN....E
A.......................E
A...O....PP......QQ....E
A...O....PP......QQ....E
A...OO...PP......QQ...E
A.......................E
A..RR....SS......TT....E
A.RR......SS....TT.....E
A.......................E
A..UU....VV......WW....E
A..UU....VV......WW....E
A..U......V......W.....E
A..U......V......W.....E
AAAAAAAAAAAAAAAAAAAAAAAA
4
```

```
........................
..AAAAA........BBBBB....
........................
..C............D........
..C............D........
..CCC..........DDD......
........................
..EEEE.........FFFF.....
.....E.........F........
........................
..GG...........HH.......
...G...........H........
...GG..........HH.......
........................
..I............J........
..III..........JJJ......
..I............J........
........................
..KK...........LL.......
..K............L........
..KK...........LL.......
........................
........................
........................
5
```

Output

1. For each test case **only** print the number of unique shapes found.
2. At the end of your program print the word **done** on a separate line.

```
5
7
done
```

5. Potholes

As winter fades, a new crop of potholes spring forth from the Earth. As usual, the municipal budget is insufficient to fix them all. Your job is to write a program to fix the most dangerous ones.

Input

1. The number of test cases will be on a line by itself.
2. Each test case:
   (a) 1 line with the number of potholes and the money available to fix them.
   (b) each pothole, on its own line, will have a two letter designation, the cost to fix it, and how dangerous it is, both as integers

```
2
5 30
PH 12 80
XJ 10 60
QR 9 50
ZA 5 300
BK 7 45
8 50
AA 12 45
B2 15 60
CK 8 30
D0 20 90
EF 5 25
G9 10 35
HX 7 28
IZ 14 55
```

Output

1. For each test case print the number of holes filled followed by the total danger reduced.
2. For each hole filled print its designation on a separate line, using the same ordering as the input.
3. At the end of your program print the word **done** on a separate line.

```
3 440
PH
XJ
ZA
4 210
B2
D0
EF
G9
done
```

6. Embedded Words

Background Information:

Given a string of letters, can a person find a subsequence of letters in index-ascending order that form a target word w?

- The letters do not have to be immediately consecutive;
- however, there may be multiple subsequences of a target word within a given string s.

Your program must find the total number of subsequences that form a target word w within an input string s.

Input

1. The string **s** to search on a line
2. followed by a target word **w** on the next line.
3. All inputs are capitalized letters.
4. The string consisting of only a period "." signifies the end of input.

```
CCCAAATTTAAA
CAT
CCCAAATTTAAA
DOG
CATCATCATCAT
CAT
.
```

Output

1. The number of subsequences of **w** found within **s**.
2. At the end of your program print the word **done** on a separate line.

```
27
0
20
done
```

7. Hip and Hippie Numbers

Background Information:

A number is "hip" if it's not a prime number and its prime decomposition contains no repeated factors which means that it's square-free. For example, 30, 77, and 561 are hip because their factorizations do not require any prime to be used more than once.

- We have $30 = 2 \times 3 \times 5$,
- $77 = 11 \times 7$, and
- $561 = 3 \times 11 \times 17$.

Note that you are competing in a hip contest.

The first non-hip or square numbers are 4, 8, 9, 12, 16, and 18.

A number N is a hippie number if it's hip and N - 1 is divisible by one less than each of the prime factors of N. For example, 561 is a hippie number because it's hip and 560 is divisible by 2, 10, and 16.

Write a program that inputs a range and outputs all of the hippie numbers in this range.

Input

1. Integers A and B with $2 \leq A \leq 1,000,000$ and $A \leq B \leq 1,000,000$ each on a separate line.
2. -1 indicates the end of input.

```
1000
2000
4
5
-1
```

Output

1. In ascending order, all of the hippie numbers in the given range, inclusive of A and B, one per line.
2. If no hippie numbers are found, print the word **None** on a separate line.
3. At the end of your program print the word **done** on a separate line.

```
1105
1729
None
done
```