acm

# ProgrammingJam 6.0

جامعة الأمير سلطان

PRINCE SULTAN
UNIVERSITY

PSU.1998

# Getting-Started-Jam6

Asalamalikum WarahmatuAllah Wabarakatuhu

The Programming Jam 6.0 team strongly welcomes you to this years competition.

This year, the competition is completely virtual due to COVID-19 restrictions in place in many areas of the Kingdom.

We pray that you and your families stay safe and stay healthy.

The submission process with be different compared to the ACM ICPC competitions. All teams are allowed to use Internet resources, however strict monitoring would be in place for any illegal activity on the forum. All submissions would also be checked against any plagiarism. The Programming Jam team have tried their best to keep the order/type of questions as close as possible to the ACM ICPC format. There are 18 problems to be solved in 9 hours. Each problem has some points associated with it. The team earning the highest score will win this competition.

Happy hunting for points.

**Input Format**

Your program reads a line from the standard Input (Console)

**Constraints**

String length should not exceed 100 characters

**Output Format**

The program displays the same line on Standard Output (Console)

# Hack the sequence

Ahmed is learning to be a cyber hacker. He writes a program that sniffs the network traffic for meaningful phrases. He knows that the organization that he is trying to hack, requires all of their employees to send secret messages by using a special sequence. This sequence changes every day!

Ahmed thinks that if he can determine the secret sequence, he can read the message and hack the communication.

Today is Ahmed's lucky day. He learned that the sequence is constructed using multiples of integers, in which every number is the multiplication of the previous two numbers.

Help Ahmed write this program.

### Input Format

A valid string should contain at least three digits to make one sequence

### Constraints

A String of input consisting of digits.

### Output Format

The output will be “True” if the string is a sequence of multiples; “False”, if the string is not a sequence of multiples, “Invalid” if the sequence has less than three digits.

### Sample Input 0

```
133927
```

### Sample Output 0

```
True
```

### Explanation 0

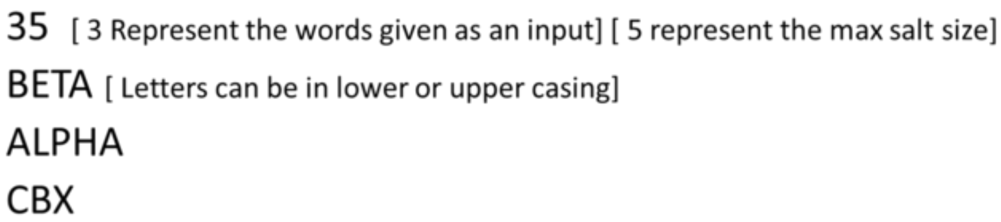
In the sample input:

```
1*3=3, 3*3=9, 3*9=27
```

So a valid sequence is found, hence True

# Ceaser Majnoon

You task is to create a new cipher that is based on Ceaser cipher approach. The algoirthm works as follows 1. Given input includes number of words and salt size



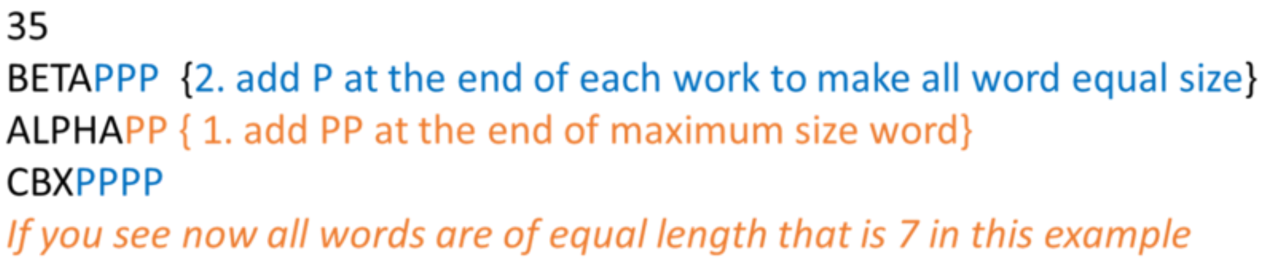
35 [ 3 Represent the words given as an input] [ 5 represent the max salt size]

BETA [ Letters can be in lower or upper casing]

ALPHA

CBX

2. The first task is to find the max size of string, and add PP to the maximum while complete other words with adding P. The complete size is given as below



35

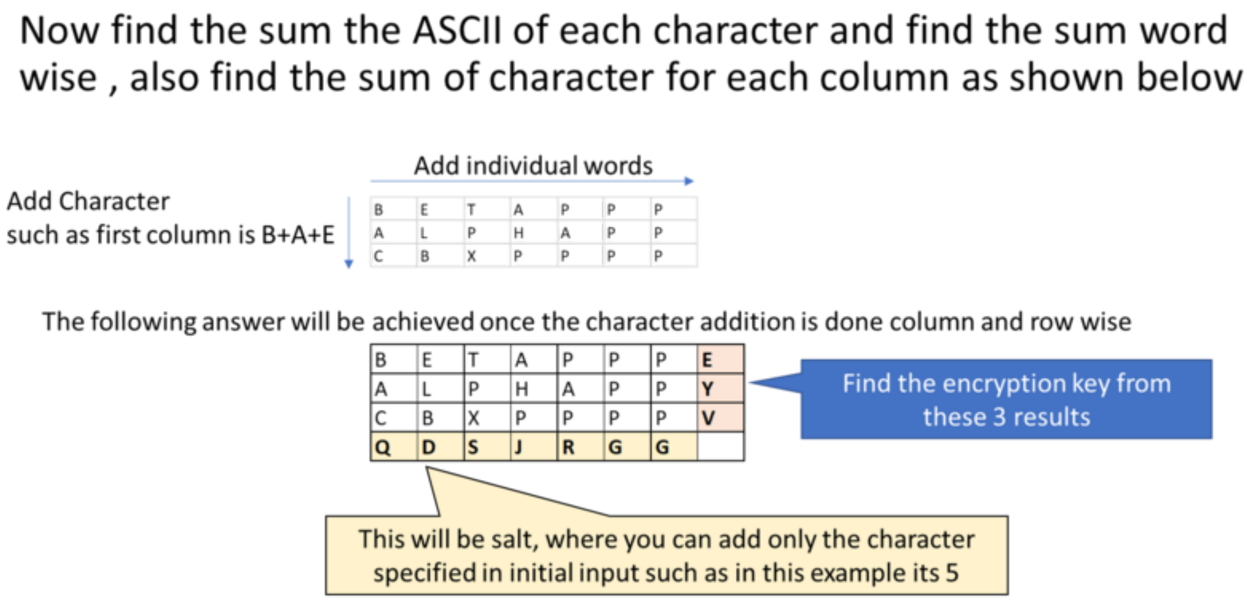
BETAPPP {2. add P at the end of each work to make all word equal size}

ALPHAPP { 1. add PP at the end of maximum size word}

CBXPPPP

*If you see now all words are of equal length that is 7 in this example*

3. Find the ASCII and its sum on each word and column wise



Now find the sum the ASCII of each character and find the sum word wise , also find the sum of character for each column as shown below

Add individual words

Add Character such as first column is B+A+E

| | | | | | | |
|---|---|---|---|---|---|---|
| B | E | T | A | P | P | P |
| A | L | P | H | A | P | P |
| C | B | X | P | P | P | P |

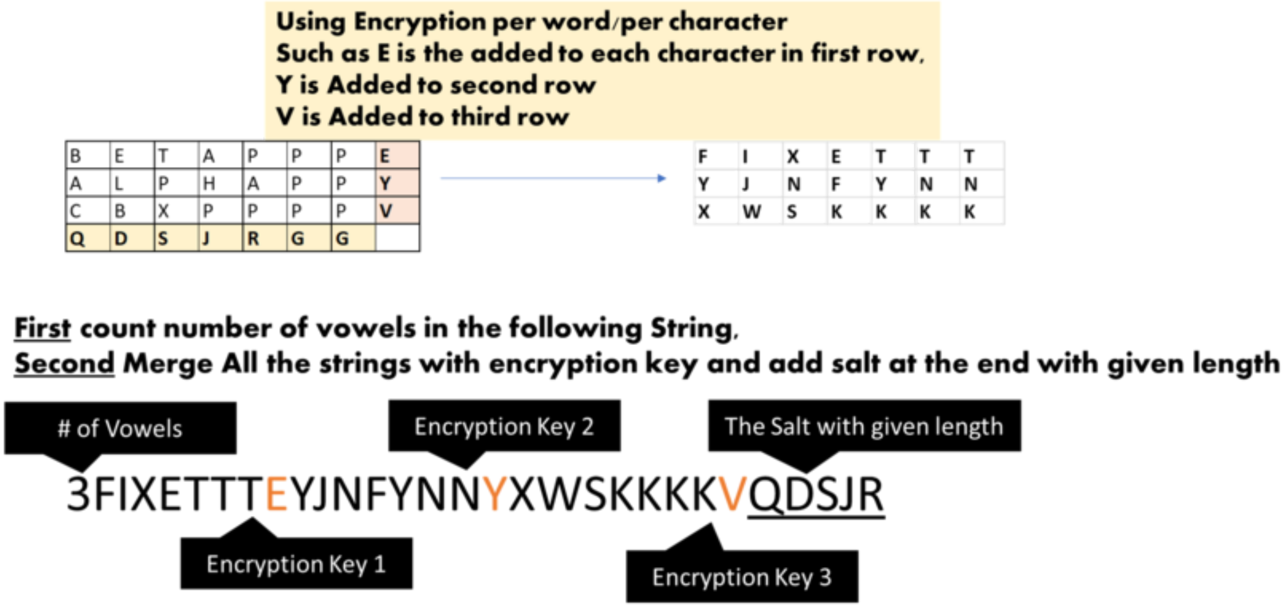
The following answer will be achieved once the character addition is done column and row wise

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | E | T | A | P | P | P | **E** |
| A | L | P | H | A | P | P | **Y** |
| C | B | X | P | P | P | P | **V** |
| **Q** | **D** | **S** | **J** | **R** | **G** | **G** | |

Find the encryption key from these 3 results

This will be salt, where you can add only the character specified in initial input such as in this example its 5

4. Lets apply the encryption



Using Encryption per word/per character
Such as E is the added to each character in first row,
Y is Added to second row
V is Added to third row

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | E | T | A | P | P | P | E |
| A | L | P | H | A | P | P | Y |
| C | B | X | P | P | P | P | V |
| Q | D | S | J | R | G | G | |

| | | | | | | |
|---|---|---|---|---|---|---|
| F | I | X | E | T | T | T |
| Y | J | N | F | Y | N | N |
| X | W | S | K | K | K | K |

**First** count number of vowels in the following String,
**Second** Merge All the strings with encryption key and add salt at the end with given length

### Input Format

# of words + Salt Size

Words in each line

### Constraints

Numbers and special character are not allowed. Character can be in upper case or lower case letter, but they are considered as same. Max length of each word can be less than or equal to 8. Salt cannot be more than given size in input In case if there is one of constraint in input just return "**Programming JAM 6.0**"

### Output Format

# of vowels + {Encrypted word + Encryption Key } + Salt

### Sample Input 0

```
35
BETA
ALPHA
CBX
```

### Sample Output 0

```
3FIXETTTEYJNFYNNYXWSKKKKVQDSJR
```

### Explanation 0

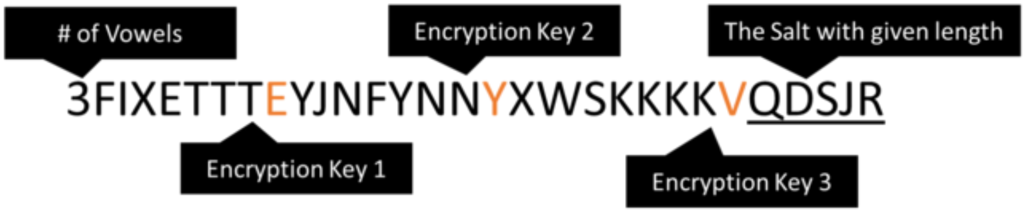
**Using Encryption per word/per character**
**Such as E is the added to each character in first row,**
**Y is Added to second row**
**V is Added to third row**

| | | | | | | | |
|---|---|---|---|---|---|---|---|
| B | E | T | A | P | P | P | **E** |
| A | L | P | H | A | P | P | **Y** |
| C | B | X | P | P | P | P | **V** |
| **Q** | **D** | **S** | **J** | **R** | **G** | **G** | |

→

| | | | | | | |
|---|---|---|---|---|---|---|
| F | I | X | E | T | T | T |
| Y | J | N | F | Y | N | N |
| X | W | S | K | K | K | K |

**<u>First</u> count number of vowels in the following String,**
**<u>Second</u> Merge All the strings with encryption key and add salt at the end with given length**



# of Vowels

Encryption Key 2

The Salt with given length

3FIXETTTEYJNFYNNYXWSKKKKVQDSJR

Encryption Key 1

Encryption Key 3

# Save-the-Pac-man

Mr. Ghost is the mortal enemy of Mr. Pacman. Ghost wishes to catch Pacman and eat him for dinner. Both live in a 2x2 grid simulation world (maze) and move in straight lines in any direction. Pacman eats food pellets and hides in safe houses which are safely installed in the maze.

In a specific situation, Mr. Pacman decides to hide in a particular safe house and starts moving in the maze towards it in a straight line at a fixed speed. However, Mr. Ghost has several secret sensors and can predict which safe house Mr. Pacman is heading to. Accordingly, Mr. Ghost moves to that house twice faster than Mr. Pacman. If Mr. Ghost reaches the safe house first, Mr. Pacman will be served as dinner to Mr. Ghost! Otherwise, Mr. Pacman will be safe and can still enjoy eating the remaining food pellets.

Use your Geometry, Calculus skills to determine if Pacman is alive or dead!

Help him live!

**Input Format**

The first line of input contain a single integer n determining the number of sets to read. The first line of each set contains one integer s and four floating point numbers. The integer s denotes how many safe houses are in the set. The four floating point numbers denote the (x_coord, y_coord) coordinates of Mr. Pacman followed by the (x_coord, y_coord) coordinates of Mr. Ghost. The subsequent s lines of input each contain two floating point numbers: the (x, y) coordinates of a safe house.

All distances are in meters to the nearest millimeter.

**Constraints**

All floating point values are to the precision 3. e.g. 1.000

0

0

### Output Format

For every set display "dead" if Pacman dies; and "alive" if he lives with the dimensions of the safe house where he is hiding.

### Sample Input 0

```
2
1 1.000 1.000 2.000 2.000
1.500 1.500
2 2.000 2.000 1.000 1.000
1.500 1.500
2.500 2.500
```

### Sample Output 0

```
dead
alive 2.500 2.500
```

### Explanation 0

There are two sets in this simulation.

The first set has only one safe house. Mr. Ghost reaches the safe-house first; hence Mr. Pacman is dead.

The second set has two safe houses. Mr. Pacman reaches safe-house (2.500, 2.500) before Mr. Ghost hence he is alive.

# Cops VS Joyriders

The cops in New York are trying to eliminate joyriding. A joyrider is someone who drives fast and dangerously for pleasure, especially in a stolen car. Given an array A of size N representing the main streets of New York, where each element consists of either a 'C' for cop or a 'J' for joyrider, you are asked to write a program to find the maximum number of joyriders that can be apprehended by the cops, with the conditions that each cop can apprehend only one joyrider, and a cop cannot apprehend a joyrider who is more than S streets away from him.

### Input Format

Each test case consists of 2 lines consisting of the number N of streets in the array, followed on the same line by the value of S allowed, followed on a new line by the contents of the array (N characters, either ‘C’ or ‘J’).

### Constraints

1 <= N <= 500
A[i] = ‘C’ or ‘J’
1 <= K <= 15

### Output Format

For each test case, output the maximum number of joyriders that can be apprehended.

### Sample Input 0

```
5 1
CJJCJ
```

### Sample Output 0

```
2
```

### Explanation 0

A maximum of 2 joyriders can be apprehended. The cop at index 0 apprehends the joyrider at index 1, and the cop at index 3 can apprehend either the joyrider at index 2 or the joyrider at index 4.

### Sample Input 1

```
6 2
JJCCJC
```

### Sample Output 1

```
3
```

### Sample Input 2

```
10 3
CJJJJCCJJC
```

### Sample Output 2

```
4
```

# Prolific-Researchers

HackerRank

At your university, many professors publish research papers working in a team. Each publication must include the Title of research, names of authors in order of contribution and other relevant information.

This year N number of researchers published M number of research papers. Many of these papers were co-authored i.e. authored by more than 1 person.

Write a program to determine which authors team published the most papers.

### Input Format

- The first line consists of N (Number of Authors) and M (Number of papers).
- The following M lines consist of M number of papers.
- Each line consists of space separated integers idenfiying the researchers who published the most papers.

### Constraints

- 0<=N<=100
- 0<=M<=100

### Output Format

One line of output is generated showing team of authors that have published the most papers. If there is a tie i.e. two or more teams of authors having the same max number of papers, show them in the order, where the results must be organized by the smallest author number appearing first.

The authors are always printed in chronological order (1 2 ...). If the input is invalid, or no output could be generated, the program prints -1 to the standard out.

### Sample Input 0

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

### Sample Output 0

```
1 2
```

### Explanation 0

N is 3, M is 3. Authors 1 and 2 published 2 papers (1 2) and (2 1). Authors 1 and 3 published 1 paper.

Hence (1 2) is output.

### Sample Input 1

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

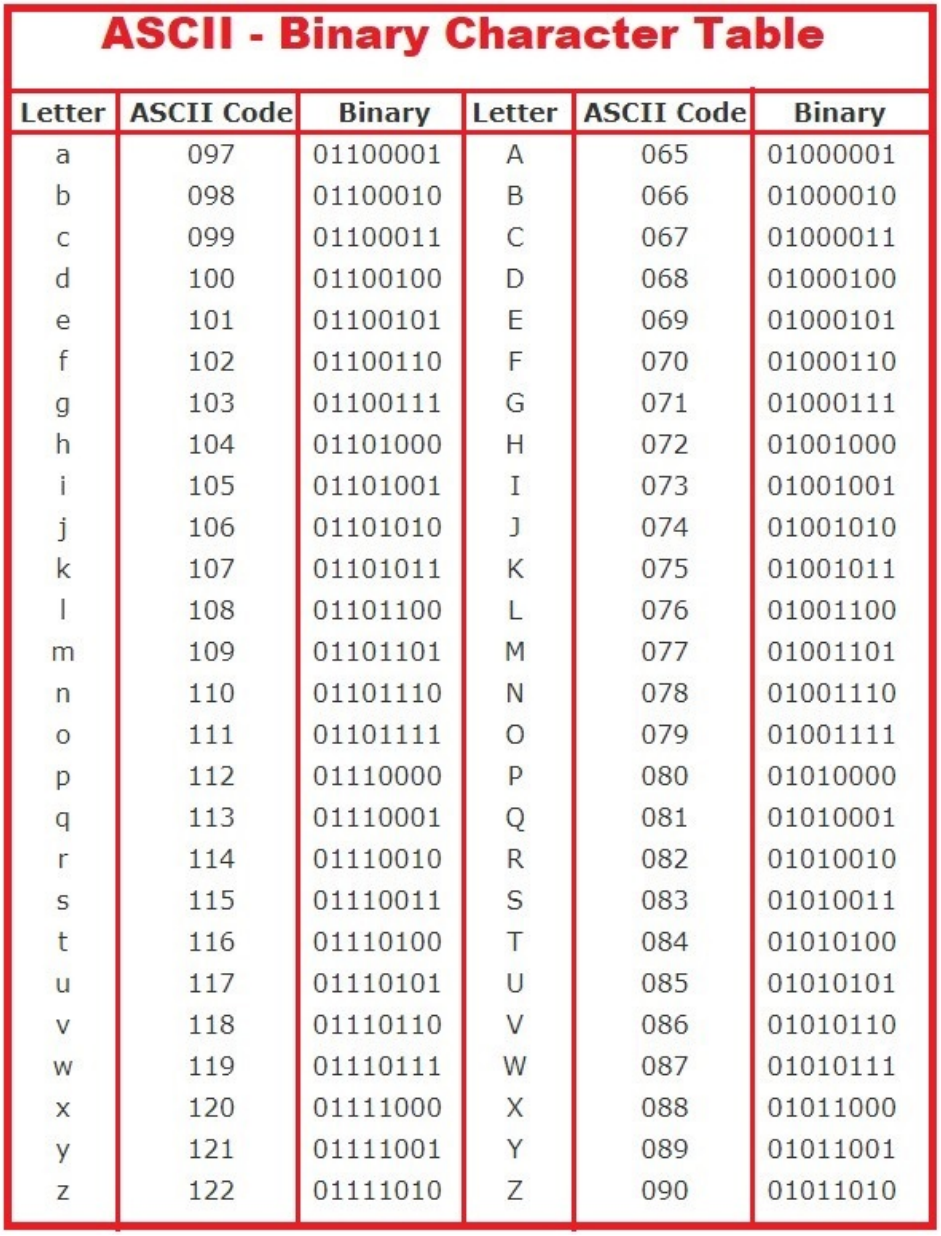
### Sample Output 1

```
1 2
3 4
```

# Hidden-Messages

Sean and John are best friends. Sean sent some pictures to John and asked him to decode the secret message hidden within. Help John find the secret messages by writing a java program.

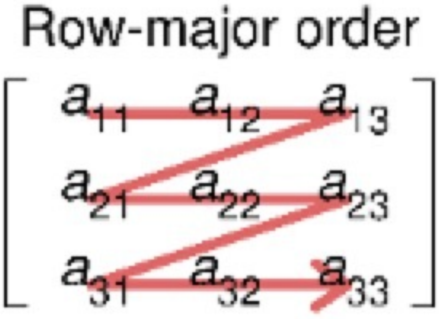
- The image is represented as a matrix of 6x4 pixels.
- The pixel is represented as an array of 8 bits.
- The hidden message is a set of characters.
- Each character is represented by its ASCII code (8 bits).



ASCII - Binary Character Table

| Letter | ASCII Code | Binary | Letter | ASCII Code | Binary |
|---|---|---|---|---|---|
| a | 097 | 01100001 | A | 065 | 01000001 |
| b | 098 | 01100010 | B | 066 | 01000010 |
| c | 099 | 01100011 | C | 067 | 01000011 |
| d | 100 | 01100100 | D | 068 | 01000100 |
| e | 101 | 01100101 | E | 069 | 01000101 |
| f | 102 | 01100110 | F | 070 | 01000110 |
| g | 103 | 01100111 | G | 071 | 01000111 |
| h | 104 | 01101000 | H | 072 | 01001000 |
| i | 105 | 01101001 | I | 073 | 01001001 |
| j | 106 | 01101010 | J | 074 | 01001010 |
| k | 107 | 01101011 | K | 075 | 01001011 |
| l | 108 | 01101100 | L | 076 | 01001100 |
| m | 109 | 01101101 | M | 077 | 01001101 |
| n | 110 | 01101110 | N | 078 | 01001110 |
| o | 111 | 01101111 | O | 079 | 01001111 |
| p | 112 | 01110000 | P | 080 | 01010000 |
| q | 113 | 01110001 | Q | 081 | 01010001 |
| r | 114 | 01110010 | R | 082 | 01010010 |
| s | 115 | 01110011 | S | 083 | 01010011 |
| t | 116 | 01110100 | T | 084 | 01010100 |
| u | 117 | 01110101 | U | 085 | 01010101 |
| v | 118 | 01110110 | V | 086 | 01010110 |
| w | 119 | 01110111 | W | 087 | 01010111 |
| x | 120 | 01111000 | X | 088 | 01011000 |
| y | 121 | 01111001 | Y | 089 | 01011001 |
| z | 122 | 01111010 | Z | 090 | 01011010 |

The hiding strategy used is known as the Least Significant Bit (LSB), which hides parts of the characters within the least significant bits of the pixels. In this problem, we use only the two LSB bits of the pixels in the Row-major Order (Z order). The character '#' marks the end of the message.


Row-major order
$a_{11}$ $a_{12}$ $a_{13}$
$a_{21}$ $a_{22}$ $a_{23}$
$a_{31}$ $a_{32}$ $a_{33}$

**Input Format**

Image in a form of a 48-character string that is the concatenation of all the image 6x4-byte matrix elements in the Z order.

**Constraints**

Input String size is always 48 characters.

**Output Format**

A String with the hidden message with size 5 characters (or less).

**Sample Input 0**

```
99D0C2CCB9F8B99589582BA8FDE4875CE180FF13C0EAA0CB
```

**Sample Output 0**

```
HELLO
```

**Explanation 0**

“99D0C2CCB9F8B99589582BA8FDE4875CE180FF13C0EAA0CB”

Which represents the following matrix of 6x4 couples of hexadecimal characters: { 99 D0 C2 CC B9 F8 B9 95 89 58 2B A8 FD E4 87 5C E1 80 FF 13 C0 EA A0 CB } Which is the equivalent of the following matrix in binary:

{10011001} {11010000} {11000010} {11001100} {10111001} {11111000}
{10111001} {10010101} {10001001} {01011000} {00101011} {10101000}
{11111101} {11100100} {10000111} {01011100} {11100001} {10000000}
{11111111} {00010011} {11000000} {11101010} {10100000} {11001011}

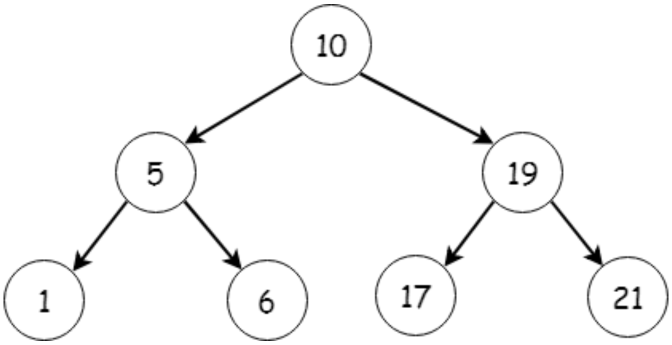
The red colored bits concatenated together form the hidden message, which will be retrieved by the recipient.

Example 01001000 map to letter H 01000101 map to letter E

and so on.

# Max-BinaryTree

Max-BinaryTree is a Binary Tree such that all the levels of the tree should have the maximum number of nodes (including the last level).



Write a program reads a sequence of integers separated by spaces ending with a new line character, representing the in-order traversal (left subtree, root, right subtree) of a complete binary tree.

Given the in-order traversal, the program should print the in-order traversal of the tree that results from adding one more level to the provided tree. The new level will be constructed by adding a left child that is one value less than the leaf node and a right child that is one value more than the right child.

If the input tree is not a Max-BinaryTree, -1 is printed to the standard output.

### Input Format

One line containing label M of type integer, seperated by spaces, representing the in-order traversal of a binary tree

### Constraints

0 <= M < 10

### Output Format

The in-order traversal of the binary tree after adding the new level if the tree is complete. If the tree is not Max-BinaryTree, -1 is printed to the standard out.

### Sample Input 0

```
3 1 4 0 5 2 6
```

### Sample Output 0

```
2 3 4 1 3 4 5 0 4 5 6 2 5 6 7
```

### Explanation 0

The following tree will be constructed with the provided input

```
         0
        / \
       1   2
      / \ / \
     3  4 5  6
```

The output will be

```
           0
        /    \
      1        2
    /   \    /   \
  3     4    5    6
 / \   / \  / \  / \
2  4   3 5  4 6  5  7
```

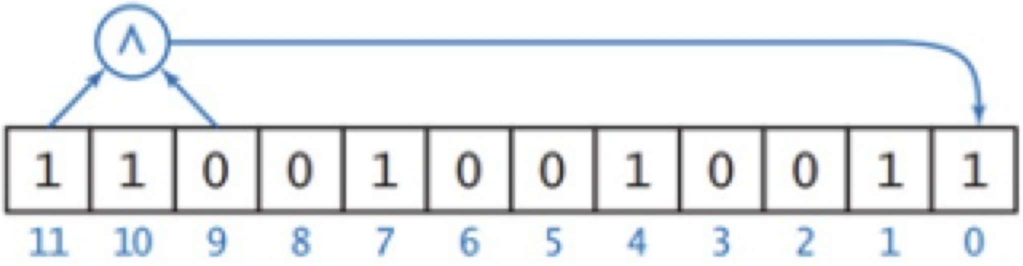
Note that a new level was added.

# Secret-Key-Generator

Sam keeps forgetting his password all the times. He went to see a Binary Guru who lives in a cave on top of the mountain. The Binary Guru was previously a programmer like you, but now he is retired and has given up on his life. He teaches sam a valuable lesson, "Stay away from computers!". Unfortunately, Sam cannot process this, he has a long life to live.

The angry Guru, noticing that Sam wont budge and leave, tells Sam a secret algorithm to generate binary keys.

Assume a 11 bit register with binary values. Repetedly XOR bit 11 and 9 to generate a secret key of length n. He draws this figure on the cave's wall.



Having learned this way to making keys, Sam's glows with happiness. All he has to do is to write a program.

Help Sam write this program.

### Input Format

A String of length 11 characters followed by an integer n.

### Constraints

0 < n < 100

String may contain only 0s and 1s.

### Output Format

A String of length n composed of 0s and 1s.

### Sample Input 0

```
11001001001 12
```

### Sample Output 0

```
110010010011
```

### Explanation 0

For this input: 11001001111 12

The 9th bit is 0 The 11th bit is 1 XOR of these two is 1 1 is appended to the end of the string: 110010011111

# Drone-Lights

Ahmed works with a company that provides drones for re-creational purposes. A very common application is to light up the sky with drones! The drones can be programmed to illuminate the lights and show/draw interesting patterns in the sky.

A customer can request N number of drones. Each drone can illuminate one light per drone. Ahmed's job is to program the drones so that lights can be displayed in a choregraphed order. Ahmed uses a control panel that has 4 switches. The function of switches is as follows:

- Switch 1: Inverse the state of all lights in all drones.
- Switch 2: Inverse only the drones in EVEN position.
- Switch 3: Inverse only the drones in ODD position.
- Switch 4: Inverse only the drones in position (3K + 1) mod N, where K is an integer.

At the begining, all lights are off (0). Ahmed plays with the switches by pressing B number of times, to create a Input vector, which creates the desired effect.

Help Ahmed write a program that provides an output giving the state of each drone (0 is off, 1 is on) as a String of 1s and 0s.

**Input Format**

Two lines consisting of Integers.

First line contains Values of N, B and k. Second line contains the Input vector of length B.

All values are space seperated.

**Constraints**

0

**Output Format**

The output contains only one line: the string of 1s and 0s written by your program.

**Sample Input 0**

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

### Sample Output 0

```
1110
```

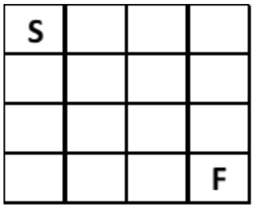
### Explanation 0

- N is 4
- B is 3, which means 3 switches are pressed in the order 1 then 2 then 4.
- K is 1
- Initially all drone-lights are 0-0-0-0
- After 1 is pressed drone-lights are 1-1-1-1
- After 2 is pressed drone-lights are 1-0-1-0
- After 4 is pressed drone-lights are 0-0-1-0

# Smart Shopper

HackerRank

Carrefour supermarket is giving away a shopping trolley full of groceries for a smart shopper. A lucky draw is organized daily among all shoppers. The shopper drawn walks through a giant-sized (N X N) board with values at each of the squares on the board. He must start at square S and finish at square F of the board, by taking steps either to the right, left, down or up (moving diagonally is not allowed) incurring the minimum total. You are asked to write a program to find the minimum total to go from start to finish.



| | | | |
|---|---|---|---|
| S | | | |
| | | | |
| | | | |
| | | | F |

### Input Format

Each test case consists of N + 1 lines. The first line containing an integer N being the size of the square board, followed by N lines, each line consisting of N integer values, separated by a single space, representing the value V on each square of the board.

### Constraints

1 <= N <= 50
0 <= V <= 9999

### Output Format

For each test case, output one single number, the minimum total to go from start to destination.

### Sample Input 0

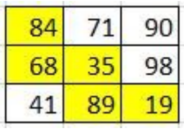
```
3
84 71 90
68 35 98
41 89 19
```

### Sample Output 0

```
295
```

### Explanation 0

The path that yields the minimum total (295) is highlighted.



| | | |
|---|---|---|
| **84** | 71 | 90 |
| **68** | **35** | 98 |
| 41 | **89** | **19** |

### Sample Input 1

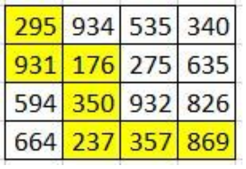
```
4
295 934 535 340
931 176 275 635
594 350 932 826
664 237 357 869
```

### Sample Output 1

```
3215
```

### Explanation 1

The path that yields the minimum total (3215) is highlighted.



| | | | |
|---|---|---|---|
| 295 | 934 | 535 | 340 |
| 931 | 176 | 275 | 635 |
| 594 | 350 | 932 | 826 |
| 664 | 237 | 357 | 869 |

### Sample Input 2

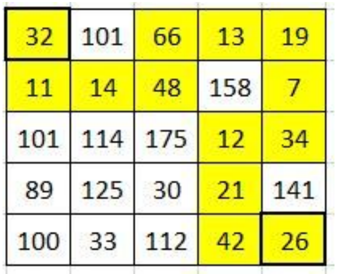
```
5
32 101 66 13 19
11 14 48 158 7
101 114 175 12 34
89 125 30 21 141
100 33 112 42 26
```

### Sample Output 2

```
345
```

### Explanation 2

The path that yields the minimum total (345) is highlighted.



| | | | | |
|---|---|---|---|---|
| 32 | 101 | 66 | 13 | 19 |
| 11 | 14 | 48 | 158 | 7 |
| 101 | 114 | 175 | 12 | 34 |
| 89 | 125 | 30 | 21 | 141 |
| 100 | 33 | 112 | 42 | 26 |

# Duplicate Detection 1

Some students are trying to outsmart their professor by copying programming assignments from their colleagues and submitting as their own work after making minor modifications. Help your professor by writing an intelligent duplicate code detection program.

Given two programs written in Sava (Simplified Java) language, you need to find out the largest chunk of duplicate code in the two files.

Sava language rules are:

1. There are 5 types of tokens: identifiers, keywords, operators, literals, and punctuators
2. Identifiers are words containing upper or lower case letters, except for keywords
3. Keywords are boolean, break, continue, do, else, final, if, int, return, void, while
4. Operators are +, -, /, *, %, =, ++, --, !, ==, >, >=, <, <=, !=,&&, ||
5. Literals are positive integers between 0 and 99999, true, false
6. Punctuators are ; , . ( ) { } [ ]
7. Tokens are always separated by a space character
8. There can be no blank lines

### Input Format

Each test case consists of two files 1 and 2, that contain code written in Sava language with n and m number of lines respectively. Your program read two integers values n and m on the first line of input. n is the number of lines to be read (as part of file 1) followed by m lines for file 2.

### Constraints

0 < n < 100 0 < m < 100

We will ignore the differences in identifiers and treat all identifiers as equal

### Output Format

For each test case, output the longest duplicate in the following format:

```
integer p, where p is Line number where the duplicate chunk starts in file 1
integer q, where q is the Token number in that line where the duplicate chunk starts
integer r, where r is the Total number of tokens in this file that are part of the duplicate chunk
```

### Sample Input 0

```
3 3
int first = 10 ;
int second = 20 ;
int sum = first + second ;
int f = 10 ;
int s = 20 ;
int r = f - s ;
```

### Sample Output 0

```
1
1
14
```

### Explanation 0

The program reads two file each of length 3 lines of code. So the first file is:

```
int first = 10 ;
int second = 20 ;
int sum = first + second ;
```

The second file is:

```
int f = 10 ;
int s = 20 ;
int r = f - s ;
```

The output is

```
1
1
14
```

We find that "int" is duplicate on line #1. so we print 1.

The first token on the first line, i.e. "int" is classed as duplicate so we print token # 1.

A total of 14 token match as duplicate, so 14 is printed.

### Sample Input 1

```
5 3
bool remainder = false ;
int dividend = 25 , divisor = 4 ;
int quotient = dividend / divisor ;
if ( dividend % divisor ) > 0 )
  remainder = true ;
int dividend = 25 , divisor = 4 ;
int quotient = dividend / divisor ;
int remainder = dividend % divisor ;
```

### Sample Output 1

```
2
1
16
```

### Explanation 1

Content of first file is

```
bool remainder = false ;
int dividend = 25 , divisor = 4 ;
int quotient = dividend / divisor ;
if ( dividend % divisor ) > 0 )
  remainder = true ;
```

Content of second file is

```
int dividend = 25 , divisor = 4 ;
int quotient = dividend / divisor ;
int remainder = dividend % divisor ;
```

The output generated is

```
2
1
16
```

Duplicate chunk is found on line 2 in file 1.

The first identifier "int" is duplicate. There are a total of 16 tokens that are duplicate.

# COVID is killing my exams!

MY university has created a temporary Exam hall to hold major/mid-term and final exams for this year. This exam hall is designed to accomodate all COVID19 precautionary requirements. All students will go to this hall at the assigned time-slots to do their exams. The problem is, there are too many exams!

Help MY university administration to write a scheduling program that will allocate the exam hall to hold exams on campus. Each exam time-slot is numbered (integer) with a starting and an ending time. The exams always start at the top of the hour (e.g. 10 am). The minimum time allocated for each slot is one hour.

To support conducting the maximum number of exams, the administration decided to use the greedy approach in choosing the time-slots with earliest ending time. If exams have identical ending time, the preference would be give to the shorter exam.

The administration hopes that this gives an optimal solution (maximum utilization of exam hall). Your program returns the maximum number of exams without any overlaps.

### Input Format

Reads integer n on the first line of input; n is the number of exams (line to be read).

Each following line contains 3 integer values No, start, and end. No is the unique exam number, start and end are the starting and ending times. All input values are integers.

### Constraints

0 <= n <= 100 0 <= No <= 100 0 <= start <24 0 <= end < 24 start < end

### Output Format

Space seperated integer values. Each value represents the exam number that can be scehduled without any conflict/overlap.

### Sample Input 0

```
3
0 10 12
1 12 13
2 15 17
```

### Sample Output 0

```
0 1 2
```

### Explanation 0

Exams 0, 1 and 2 donot overlap, hence they can be scheduled

# Inheritance 12

A old bullionaire is dying. His wish is to have only ONE of his decendants inherit ALL of his wealth, but which one? He has N number of decendants.

He works out a solution to find the one lucky child who inherits his wealth. He will get N tokens, label each token with an integer value and assign each token to a descendant. Each token gets a value only once.

He places all the tokens in a circle around him, numbered 1 to N, and starts eliminating one in K until there is only one left ...

Write a program that takes two integer values N and K and prints the token number of the lucky decendant.

### Input Format

The first line of the entry contains the two integers N and K.

### Constraints

0 < N 0 < K

### Output Format

The output contains a single integer value

### Sample Input 0

```
7 3
```

### Sample Output 0

```
4
```

### Explanation 0

There are 7 descendants. N = 7. The eliminator K is 3. In the first go, 3 is eliminated, then 6, then 2 then 7, then 5 and then 1. Only 4 is remaining.

# D-Pizza Delivery

D-Pizza has started a new branch in the professor colony. The colony has N streets numbered from 0 to N-1. The D-Pizza is located at the first street S (i.e., street no. 0) of the colony.

Initially, the manager hired one pizza delivery guy, who always chose random path for any order delivery. Later, the manager noticed frequent delays in the pizza deliveries and planned to create a map of the whole colony containing the reachability cost (i.e., travel time) of each street.

The main purpose was to discover a short path (less travel time) list from the delivery branch to all other streets. The manager initially formed a N×N matrix T, representing the travel time cost (minutes) among the streets. Where T[i][j] represents the required time to reach from ith street to the jth street, while T[i][i] = 0 indicates the delivery to the same street.

Write a program that could create a list of all minimum cost paths from delivery branch (i.e., street no. 0) to all other streets.

### Input Format

Each test case consists of an input integer N for describing the number of streets in professor colony. In addition, each test case accepts N×N integers representing the travel time of delivery guy from each ith street to jth street.

### Constraints

T[i][j] = C, where C > 0, if street i is directly connected to street j. T[i][j] = INF, if street i is not directly connected to street j. T[i][i] = 0, if the delivery to the same street.

### Output Format

For each test case, the output consists of three space-seperated columns, i.e., Street No., Travel Cost, and Path. The output represents the travel cost and path from the D-Pizza delivery branch to all other streets in separate lines.

### Sample Input 0

```
8
0 2 9 INF 7 INF INF INF
2 0 4 INF INF 8 INF INF
9 4 0 INF 4 INF 7 INF
INF INF INF 0 INF INF INF 9
7 INF 4 INF 0 INF 6 5
INF 8 INF INF INF 0 7 7
INF INF 7 INF 6 7 0 4
INF INF INF 9 5 7 4 0
```

### Sample Output 0

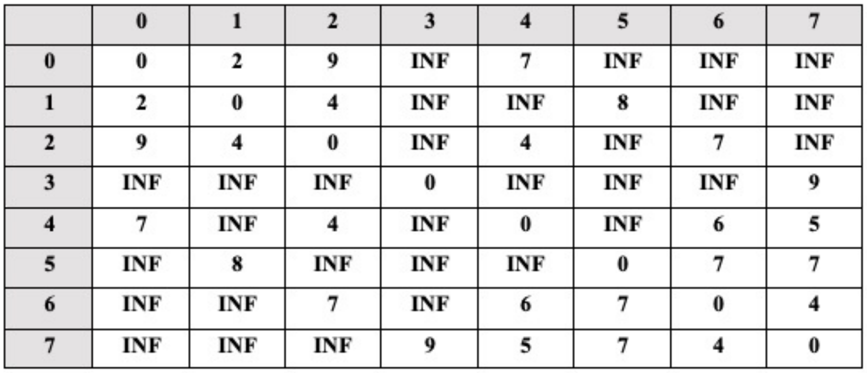
```
0 0 0
1 2 0 1
2 6 0 1 2
3 21 0 4 7 3
4 7 0 4
5 10 0 1 5
```

```
  6 13 0 1 2 6
  7 12 0 4 7
```

### Explanation 0

The above output represents the cost (travel time) and path (all intermediate streets) from the street 0 or D-Pizza delivery branch to all other streets, i.e., street [1-7].

Example: To deliver pizza from 0 to 3: Path 0 4 7 3: Cost = 7 + 5 + 9 = 21



| | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 |
|---|---|---|---|---|---|---|---|---|
| 0 | 0 | 2 | 9 | INF | 7 | INF | INF | INF |
| 1 | 2 | 0 | 4 | INF | INF | 8 | INF | INF |
| 2 | 9 | 4 | 0 | INF | 4 | INF | 7 | INF |
| 3 | INF | INF | INF | 0 | INF | INF | INF | 9 |
| 4 | 7 | INF | 4 | INF | 0 | INF | 6 | 5 |
| 5 | INF | 8 | INF | INF | INF | 0 | 7 | 7 |
| 6 | INF | INF | 7 | INF | 6 | 7 | 0 | 4 |
| 7 | INF | INF | INF | 9 | 5 | 7 | 4 | 0 |

# Tarts Galore

A tart is a baked dish consisting of a filling over a pastry base with an open top not covered with pastry. It has a circular shape and comes in three different sizes:
- Small with an area of S units$^2$ with a price of PS SAR
- Medium with an area of M units$^2$ with a price of PM SAR
- Large with an area of L units$^2$ with a price of PL SAR
Ali is a tart lover and wants for his birthday party at least X units$^2$ of strawberry tart. What is the lowest price he must pay?

### Input Format

Each test case consists of one line containing 7 integers X, S, PS, M, PM, L, PL

### Constraints

1 <= X <= 500
1 <= S <= 100
1 <= PS <= 100
1 <= M <= 100
1 <= PM <= 100
1 <= L <= 100
1 <= PL <= 100

### Output Format

For each test case, output one integer number, the lowest price in SAR that Ali must pay.

### Sample Input 0

```
21 4 55 7 155 11 305
```

### Sample Output 0

```
330
```

### Explanation 0

Ali wants 21 units$^2$ of tart.
1 Small + 1 Medium + 1 Large = 4 + 7 + 11 = 22 units$^2$ with a price of 55 + 155 + 305 = 515 SAR
2 Large = 11 + 11 = 22 units$^2$ with a price of 305 + 305 = 610 SAR
3 Medium = 7 + 7 + 7 = 21 units$^2$ with a price of 155 + 155 + 155 = 465 SAR
6 Small = 4 + 4 + 4 + 4 + 4 + 4 = 24 units$^2$ with a price of 55 + 55 + 55 + 55 + 55 + 55 = 330 SAR
etc…
Of all the combinations, 330 SAR is the lowest price.

### Sample Input 1

```
49 5 75 9 145 13 295
```

### Sample Output 1

```
745
```

### Sample Input 2

```
133 10 310 20 520 30 730
```

### Sample Output 2

```
3440
```

# Am I symmetric?

Write a program that processes a 2 dimensional matrix of Size R rows and C columns, filled with 0s and 1s describing a geometric shape. The program return 1 if there is a vertical axis of symmetry, 0 otherwise.

### Input Format

The first line contains the two integers R and C. The following R lines, each contain C integers consititing the elements of the array.

### Constraints

1 <= R <= 1000 1 <= C <= 1000

R and C are the number of rows and columns of the matrix.

### Output Format

The output contains only one line: the integer value 0 or 1.

### Sample Input 0

```
3 6
0 0 1 1 0 0
0 1 1 1 1 0
0 0 1 1 0 0
```

### Sample Output 0

```
1
```

# Mars Soccer Worldcup 2050

Its 2050 and Humans were able to establish various settelements on the planet Mars. The Mars Soccer Federation (Something like FIFA) is planning to hold the MarsCup2050. Our country United States of Marineris (USM) will send a team to the competition. Each team is composed of 5 members with an additional member as a substitute (optional).

The USM is composed of 10 states; unfortunately even in 2050, the citizens of each of these 10 states are bitter about many political tensions. The citizens do not want more than 2 citizens per state participating in the national team. The training camp can host 6 to 15 soccer players at a time. The selectors will choose a team of 5 members + 1 substitute.

Assume there are n players at the training camp. Each state is identified by a letter: A, B, C, D, E, F, G, H, I and J. A player is identified by a letter, representing the state he is from, and a unique two digit number, for example, A23, H01 etc.

Help the selectors identify the number of teams with 5 and 6 players.

### Input Format

A single line consisting of comma seperated values. Each value is composed of a character and 2 digits representing a player.

### Constraints

6<=n<=15

### Output Format

Two integer values given on two lines. The first value is number of teams with 5 members. The second value is the number of teams with 6 members.

### Sample Input 0

```
A23,C01,C02,F05,G11,B10
```

### Sample Output 0

```
2
0
```

### Explanation 0

If there are six players (n = 6): A23, C01, C02, F05, G11, B10, the possible number of teams with five players is 2, i.e.

(A23, C01, F05, G11, B10) and (A23, C02, F05, G11, B10).

The possible number of teams with six players is 0.