

**TECHMAHINDRA 2024
ASKED CODING QUESTION**



HIRING HUSTLE

DO “FOLLOW” FOR MORE

**Tech Mahindra 2024 asked
Coding Questions @hiringhustle**

Question

An e-commerce company is planning to give a special discount on all its products to its customers for the holiday. The company possesses data on its stock of N product types. The data for each product type represents the count of customers who have ordered the given product. If the data K is positive then it shows that the product has been ordered by K customers and is in stock. If the data K is negative then it shows that it has been ordered by K customers but is not in stock. The company will fulfill the order directly if the ordered product is in stock. If it is not in stock, then the company will fulfill the order after they replenish the stock from the warehouse. They are planning to offer a discount amount A for each product. The discount value will be distributed to the customers who have purchased that selected product. The discount will be distributed only if the decided amount A can be divided by the number of orders for a particular product.

Write an algorithm for the sales team to find the number of products out of N for which the discount will be distributed.

Input

The first line of the input consists of

```
1
2
3
4
5 def noOf
6     # W
7
8     ret
9
10 def mai
11     #ir
12     or
13     or
14     or
15
16     #i
17     di
18
19
20     re
21     pr
22
23 if
24     m
```


Question

Input

The first line of the input consists of an integer - *numOfProducts*, representing the number of different types of products (N).

The second line consists of N space-separated integers - *order₀*,

order₁, ..., *order_{N-1}*, representing the current status of the stock for the orders of the respective product types.

The last line consists of an integer - *disAmount*, representing the discount amount that will be distributed among the customers.

Output

Print an integer representing the number of products out of N for which the discount will be distributed.

Constraints

$$0 \leq \text{numOfProducts}, \text{disAmount} \leq 10^5$$

$$-10^6 \leq \text{order}_i \leq 10^6$$

$$0 \leq i < \text{numOfProducts}$$

Example

Input:

7

9 -13 8 -7 -8 18 10

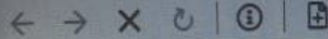
18

Output:

```
1
2
3
4
5 def numOfProducts:
6     # Write
7
8     return
9
10 def main():
11     #input
12     order =
13     order_s
14     order =
15
16     #input
17     disAmount
18
19
20     result
21     print(
22
23     if __name__ == '__main__':
24         main()
```

2)

SHL



SHL

Question

Input

The first line of the input consists of a positive integer - $base_1$, representing the base of the first triangle.

The second line consists of a positive integer - $height_1$, representing the height of the first triangle.

The third line consists of a positive integer - $base_2$ representing the base of the second triangle.

The last line consists of a positive integer - $height_2$, representing the height of the second triangle.

Output

Print a real number representing the area of the largest triangle rounded up to one decimal place.

Constraints

$0 \leq base_1, height_1, base_2, height_2 \leq$

10^9

Example

Input:

5

8

4

11

Output:

22.0

```

1 //Header Files
2 #include<stdio.h>
3 #include<stdlib.h>
4 #include<string.h>
5 #include<stdbool.h>
6
7 /* only used in st
8 typedef struct Str
9 struct String
10 {
11     char *str;
12 };
13
14 char *input(FILE *
15 {
16     int actual_size
17     char *str = (c
18     char ch;
19     if(has_space ==
20     {
21         while(EOF !
22         {
23             str[act
24             actual_
25             if(actu
26             {
27                 str
28             }
29         }
30     }
31     else
32     }

```


SHL

Question

Jackson, a math research student, is developing an application on prime numbers. For the given two integers on the display of the application, the user has to identify all the prime numbers within the given range (including the given values). Afterward, the application will sum all these prime numbers. Jackson has to write an algorithm to find the sum of all the prime numbers of the given range.

Write an algorithm to find the sum of all the prime numbers of the given range.

Input

The first line of the input consists of an integer - *rangeLeft*, representing the minimum boundary value of the given range(including the given values).

The second line consists of an integer - *rangeRight*, representing the maximum boundary value of the given range(including the given values).

Output

Print an integer representing the sum of all the prime numbers of the given range.

Constraints

$$-10^9 \leq \text{rangeLeft} < \text{rangeRight} \leq 10^9$$

Example

Input:

-30

```
1 //Header Files
2 #include<stdio.h>
3 #include<stdlib.h>
4 #include<string.h>
5 #include<stdbool.h>
6
7 /* only used in string related operations */
8 typedef struct String string;
9 struct String
10 {
11     char *str;
12 };
13
14 char *input(FILE *fp, int size, int has_space)
15 {
16     int actual_size = 0;
17     char *str = (char *)malloc(sizeof(char) * size);
18     char ch;
19     if(has_space == 1)
20     {
21         while(EOF != (ch=fgetc(fp)) && ch != '\n')
22         {
23             str[actual_size] = ch;
24             actual_size++;
25             if(actual_size >= size)
26             {
27                 str = realloc(str, size * 2);
28             }
29         }
30     }
31     else
32     {
33         while(EOF != (ch=fgetc(fp)) && ch != '\n')
34         {
35             str[actual_size] = ch;
36             actual_size++;
37             if(actual_size >= size)
```


Question

A stock trader trades in N selected stocks. The trader has calculated the relative stock price changes in the N stocks from the previous day stock prices. The lucky number of the trader is K , so the trader wishes to invest in the particular stock that has the K^{th} smallest relative stock value.

Write an algorithm for the trader to find the K^{th} smallest stock price out of the selected N stocks.

Input

The first line of the input consists of an integer - *stock_size*, representing the number of selected stocks (N). The second line consists of N space-separated integers - *stock₀*, *stock₁*, ..., *stock_{N-1}*, representing the relative stock prices of the selected stocks. The third line consists of an integer - *valueK*, representing the value K for which the trader wishes to find the stock price.

Output

Print an integer representing the K^{th} smallest stock price of selected N stocks.

Constraints

$$0 < \text{valueK} \leq \text{stock_size} \leq 10^6$$

$$0 \leq \text{stock}_i \leq 10^6$$

$$0 \leq i < \text{stock_size}$$

Example

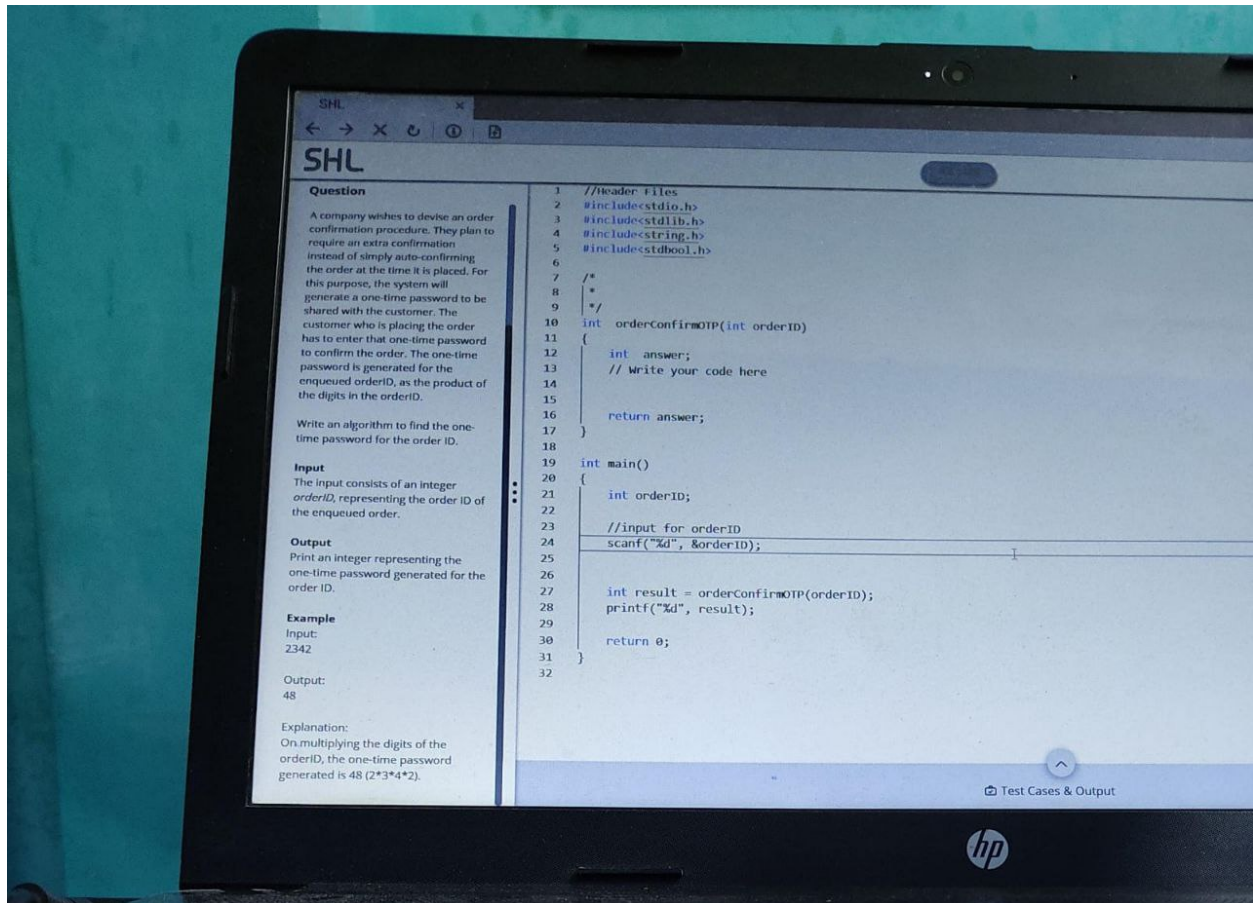
Input:

5

10 5 7 88 40

```
1
2
3
4
5 def smallestStockPrice(stock, valueK):
6     # Write your code here
7
8     return
9
10 def main():
11     # input for stock
12     stock = []
13     stock_size = int(input())
14     stock = list(map(int, input().split))
15
16     # input for valueK
17     valueK = int(input())
18
19
20     result = smallestStockPrice(stock, valueK)
21     print(result)
22
23 if __name__ == "__main__":
24     main()
```

5



6

version of Python being used is 3.10.0

A network protocol specifies how data is exchanged via transmission media. The protocol converts each message into a stream of 1s and 0s. Given a decimal number, write an algorithm to convert the number into a binary form.

Input

The input consists of an integer *num*, representing the decimal number.

Output

Print an integer representing the binary form of the given decimal number.

Example

Input:
25

Output:
11001

Explanation:

Divide the given number 25 by 2. The quotient is 12 and the remainder is 1, the least significant bit.

Now divide the quotient 12 by 2. The quotient is 6 and the remainder is 0.

6

7

8

9

10

11

12

13

14

15

16

17

18

19



The current selected programming language is C. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use `printf()` to debug your code. The `printf()` may not work in case of syntax/runtime error. The version of GCC being used is **5.5.0**.

An e-commerce company plans to give their customers a discount. The discount will be calculated on the basis of the bill amount of the order placed. The discount amount is the sum of all the odd digits in the customer's total bill amount. If no odd digit is present in the bill amount, then the discount will be zero.

Write an algorithm to find the discount for the given total bill amount.

Input

The input consists of an integer - `billAmount`, representing the customer's total bill amount.

Output

Print an integer representing the discount for the given total bill amount.

```
2  #in
3  #in
4  #in
5  #inc
6
7  /* o
8  type
9  struc
10 {
11 | c
12 };
13
14 char *
15 {
16 in
17 cha
18 cha
19 if(
20 {
21
22
23
24
25
26
27
28
29
30 }
31 else
32 }
```


Output

Print an integer representing the discount for the given total bill amount.

Constraints

$$0 < \text{billAmount} \leq 10^9$$

Example

Input:

2514795

Output:

27

Explanation:

Odd digits in the given number 2514795 are 5, 1, 7, 9, 5. The sum of these odd digits is 27.

So, the output is 27.

The current selected programming language is C. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use `printf()` to debug your code. The `printf()` may not work in case of syntax/runtime error. The version of GCC being used is **5.5.0**.

A company called Dictory is launching a new dictionary application for mobile users. Initially, the dictionary will not have any words. Instead it will be an auto-learning application that will learn according to a user's given text. When a user types text, the application auto-detects the words that appear more than once. The application then stores these words in the dictionary and uses them as suggestions in future typing sessions.

Write an algorithm to identify which words will be saved in the dictionary.

Input

The input consists of a string - `textInput`, representing the text that is given as an input to the application by the user.

```
1  #include
2
3  #include
4  #include
5  #include
6
7  /* only
8  typedef
9  struct S
10 {
11     char
12 };
13
14 char *inp
15 {
16     int ac
17     char *
18     char c
19     if(has
20     {
21         whi
22     {
23
24
25
26
27
28
29
30     }
31     else
32 }
```


It contains lowercase and uppercase English alphabets[i.e. a-z, A-Z].

Example

Input:

cat batman latt matter cat matter cat

Output:

cat matter

Explanation:

The word "cat" is repeated three times and the word "matter" is repeated two times in the text. So, the dictionary will store ["cat", "matter"].

esc

f1



f2



Question

Input

The input consists of a string - *textInput*, representing the text that is given as an input to the application by the user.

Output

Print space-separated strings in the lexicographically sorted order representing the number of repeated words detected in the input text and if no word is repeated print "NA".

Note

A word is an alphabetic sequence of characters having no whitespace and there is no punctuation in the input text.

textInput is case-sensitive (i.e. cat and CAT are considered as different words not same).

It contains lowercase and uppercase English alphabets [i.e. a-z, A-Z].

Example

Input:
cat batman latt matter cat matter cat

Output:
cat matter

Explanation

The word "cat" is repeated three times and the word "matter" is repeated twice.

```
1 //He
2 #inc
3 #inc
4 #inc
5 #inc
6
7 /* on
8 typed
9 struct
10 {
11     ch
12 };
13
14 char *i
15 {
16     int
17     char
18     char
19     if(h
20     {
21         w
22         {
23
24
25
26
27
28
29     }
30 }
31 else
32 }
```


Question

code. Once **submitted**, you cannot review this problem again. You can use `printf()` to debug your code. The `printf()` may not work in case of syntax/runtime error. The version of **GCC** being used is **5.5.0**.

The company has transmitted its data to the target server successfully. Now, on the new server, the transmitted binary data has to be stored in a user-readable format. As a developer, you have been assigned the task of converting the received binary data to user-readable decimal data.

Write an algorithm to convert the given data to the decimal format.

Input

The input consists of an integer- *data*, representing the binary form of the data received.

Output

Print an integer representing the decimal form of the received data.

Example

Input:
1001111001

Output:
633

Explanation:
On converting the given binary data, the output is 633.

Question

The current selected programming language is **Java**. We emphasize the submission of a fully working code over partially correct but efficient code. Once **submitted**, you cannot review this problem again. You can use `System.out.println()` to debug your code. The `System.out.println()` may not work in case of syntax/runtime error. The version of **JDK** being used is **1.8**.

Note: The main class name must be **"Solution"**.

Jackson, a math research student, is developing an application on prime numbers. For the given two integers on the display of the application, the user has to identify all the prime numbers within the given range (including the given values). Afterward, the application will sum all these prime numbers. Jackson has to write an algorithm to find the sum of all the prime numbers of the given range.

Write an algorithm to find the sum of all the prime numbers of the given range.

Input

The first line of the input consists of an integer - `rangeLeft`, representing the minimum boundary value of the given range (including the given values).

The second line consists of an integer - `rangeRight`, representing

```
1 import java.util.*;
2 import java.lang.*;
3 import java.io.*;
4
5 /*
6  *
7  */
8 public class Solution
9 {
10     public static int sumPrime(int rangeLeft, int rangeRight)
11     {
12         int answer = 0;
13         // Write your code here
14
15
16         return answer;
17     }
18
19     public static void main(String[] args)
20     {
21         Scanner in = new Scanner(System.in);
22         // input for rangeLeft
23         int rangeLeft = in.nextInt();
24
25         // input for rangeRight
26         int rangeRight = in.nextInt();
27
28
29         int result = sumPrime(rangeLeft, rangeRight);
30         System.out.print(result);
31
32     }
33 }
34
```

Test

Input

The first line of the input consists of an integer - *rangeLeft*, representing the minimum boundary value of the given range(including the given values).

The second line consists of an integer - *rangeRight*, representing the maximum boundary value of the given range(including the given values).

Output

Print an integer representing the sum of all the prime numbers of the given range.

Constraints

$-10^9 \leq \text{rangeLeft} < \text{rangeRight} \leq 10^9$

Example

Input:

-30

-11

Output:

-112

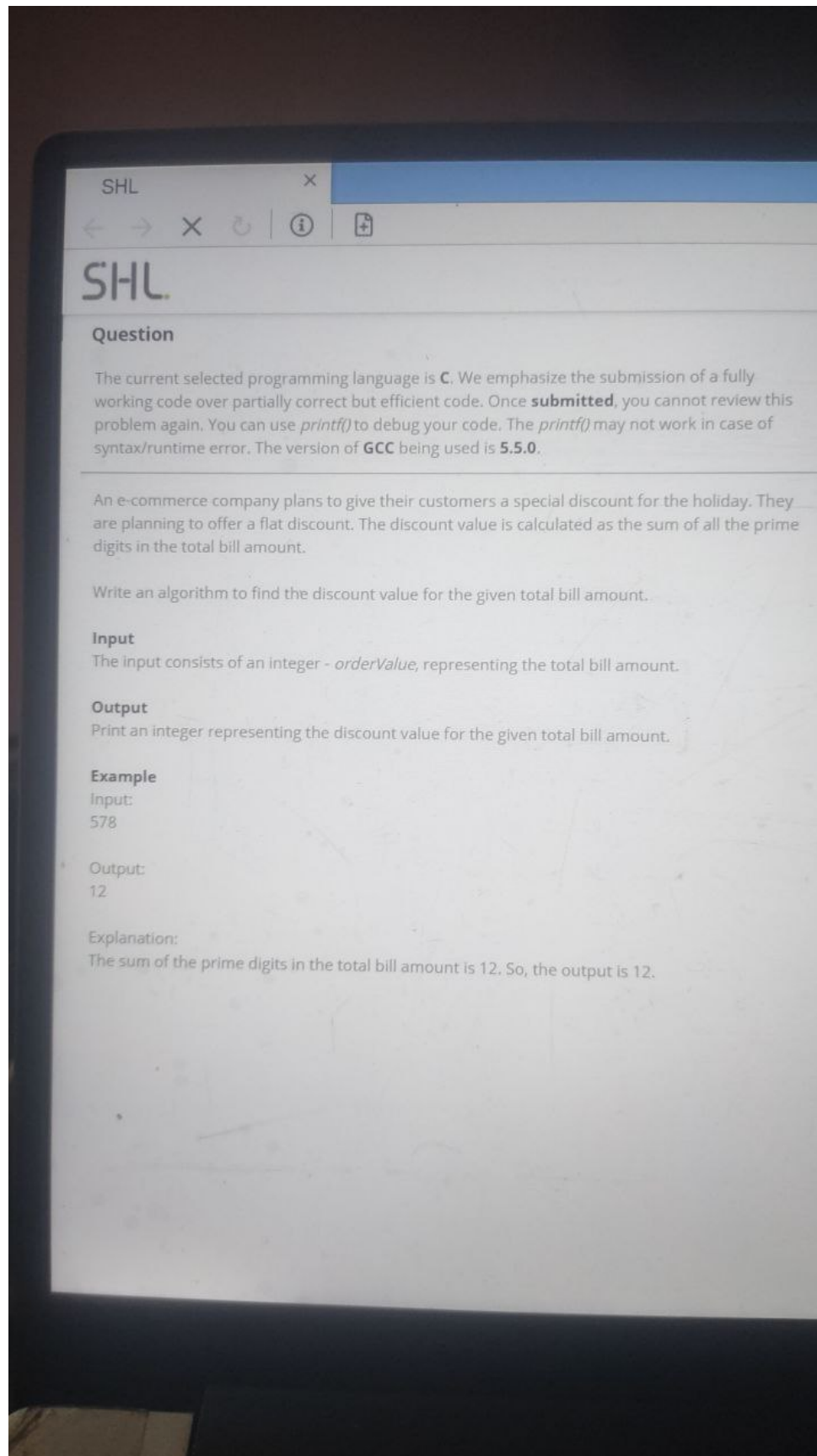
Explanation:

The prime numbers within the given range are: -29, -23, -19, -17, -13 and -11. The sum of the prime numbers is -112.

So, the output is -112.

```
16     return answer;
17 }
18
19 public static void main(String[] args)
20 {
21     Scanner in = new Scanner(System.in);
22     // input for rangeLeft
23     int rangeLeft = in.nextInt();
24
25     // input for rangeRight
26     int rangeRight = in.nextInt();
27
28
29     int result = sumPrime(rangeLeft, rangeRight);
30     System.out.print(result);
31 }
32
33 }
34
```

Test Cases



12

developing an application on prime numbers. For the given two integers on the display of the application, the user has to identify all the prime numbers within the given range (including the given values). Afterward, the application will sum all these prime numbers. Jackson has to write an algorithm to find the sum of all the prime numbers of the given range.

Write an algorithm to find the sum of all the prime numbers of the given range.

Input
The first line of the input consists of an integer - *rangeLeft*, representing the minimum boundary value of the given range(including the given values). The second line consists of an integer - *rangeRight*, representing the maximum boundary value of the given range(including the given values).

Output
Print an integer representing the sum of all the prime numbers of the given range.

Constraints
 $-10^9 \leq \text{rangeLeft} < \text{rangeRight} \leq 10^9$

```

4 #include <string.h>
5 #include <stdbool.h>
6
7 /* only used in string related operations */
8 typedef struct String string;
9 struct String
10 {
11     char *str;
12 };
13
14 char *input(FILE *fp, int size, int has_space)
15 {
16     int actual_size = 0;
17     char *str = (char *)malloc(sizeof(char)*(size+actual_size));
18     char ch;
19     if(has_space == 1)
20     {
21         while(EOF != (ch=fgetc(fp)) && ch != '\n')
22         {
23             str[actual_size] = ch;
24             actual_size++;
25             if(actual_size >= size)
26             {
27                 str = realloc(str, sizeof(char)*actual_size);
28             }
29         }
30     }
31     else
32     {
33         while(EOF != (ch=fgetc(fp)) && ch != '\n' && ch != ' ')
34         {
35             str[actual_size] = ch;
36             actual_size++;
37             if(actual_size >= size)

```

Test Cases & Output

given range(including the given values).

Output
Print an integer representing the sum of all the prime numbers of the given range.

Constraints
 $-10^9 \leq \text{rangeLeft} < \text{rangeRight} \leq 10^9$

Example
Input:
-30
-11

Output:
-112

Explanation:
The prime numbers within the given range are: -29, -23, -19, -17, -13 and -11. The sum of the prime numbers is -112.
So, the output is -112.

```

16 int actual_size = 0;
17 char *str = (char *)malloc(sizeof(char)*(size+actual_size));
18 char ch;
19 if(has_space == 1)
20 {
21     while(EOF != (ch=fgetc(fp)) && ch != '\n')
22     {
23         str[actual_size] = ch;
24         actual_size++;
25         if(actual_size >= size)
26         {
27             str = realloc(str, sizeof(char)*actual_size);
28         }
29     }
30 }
31 else
32 {
33     while(EOF != (ch=fgetc(fp)) && ch != '\n' && ch != ' ')
34     {
35         str[actual_size] = ch;
36         actual_size++;
37         if(actual_size >= size)

```

Test Cases & Output

13

again. You can use `print` to debug your code. The `print` may not work in case of syntax/runtime error. The version of Python being used is **3.5.2**

The company Digital Secure Data Transfer Solutions provides data encryption and data sharing services. Their process uses a key K for encryption when transmitting a number. To encrypt a number, each digit in the number is replaced by the K th digit after it in the number. The series of digits is considered in a cyclic fashion for the last K digits.

Write an algorithm to find the encrypted number.

Input

The first line of the input consists of an integer *data* representing the number.

The second line consists of an integer *and key*, representing the key (K).

Output

Print an integer representing the encrypted number.

Constraints

$0 < data \leq 10^9$

```
10
11
12
13
14
15
16 def ma
17     #i
18     da
19
20     #i
21     ke
22
23
24
25
```

Console Out

Case 1

Input:

Output:

Expected Outp

Console Outp

Write an algorithm to find the encrypted number.

Input

The first line of the input consists of an integer *data* representing the number.

The second line consists of an integer *and key*, representing the key (K).

Output

Print an integer representing the encrypted number.

Constraints

$$0 < data \leq 10^9$$

Example

Input:

25143

3

Output:

43251

Explanation:

Replace 2 with 4, 5 with 3, 1 with 2, 4 with 5, and 3 with 1.

So, the output is 43251.

```
21 key =
22
23
24 resul
25 print
26
27 if __nam
28 main
```

Console Output

Case 1



Input:

Output:

Expected Output

Console Output