MINDTREE AUTOMATA-17/09/19

PROBLEM-1-1:AUTO

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A company wishes to provide cab service for their N employees.The employees have IDs ranging from 0to N-1. The company has calculated the total distance from an employee’s residance to the company,considering the path to be followed by the cab is a stright path.The distance of the company from itself is 0.The distance for the employees who live to the left side of the company is represented with a negative sign.The distance for the employees who live to the right side of the company is represented with a positive sign.The cab will be alloted a range of distance.The company wishes to find the IDs of the employees who live within the particular distance range.

Write an algoritham to find the employee IDs who live within the distance range.

Input

The first line of the input consists of three space-separated integers-num,start and representing the size of the list (N);the starting value of the range,and the ending value of the range,respectively.

The second line of the input consists of N space-separated integers representing the distance of the employees from the company.

Output

Print space-separated integers representing the IDs of the employees whose distance lies within the given range else return-1.

Example

Input

6 30 50

29 38 12 48 39 55

Output

134

Explanation:

There are three employees with IDs 1,3 and 4 whose distance from the office lies within the given range.

Code:

#include <stdio.h>

int main()

{

int n,start,end,a[100],i;

scanf("%d %d %d\n",&n,&start,&end);

for(i=0;i<n;i++){

scanf("%d",&a[i]);

}

for(i=0;i<n;i++){

if(a[i]>start&&a[i]<end){

printf("%d",i);

}

}

return 0;

}

PROBLEM-1-2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Mr.Woods,an electrician has made some faulty connections of eight street lights in Timberland city.The connections are such that if the street lights adjacent to a particular light are both ON(represented as 1) or are both OFF (represented as 0),then that street light goes OFF the next night.Otherwise,it remains ON the next night.The two street lights at the end of the road have only a single adjacent street light,so the other adjacent light can be assumed to be always OFF.The state of the lights on a particular day is considered for the next day and not for the same day.

Due to this fault,the people of the city are facing difficulty in driving on the road at night.So,they have filed a complaint about this to the Head of the Federal Highway Administration.Based on this complaint the head has asked for the report of the state of street lights after M days.

Write an algoritham to output the state of the street lights after the given M days.

Input

The first line of the input consists of an integer-days, representing the number of days (M).

The next line consists of eight space-separated integers representing the current state of street lights i.e.eighter 0 or 1.

Output

Print eight space-separated integers representing the state of street lights after M days.

Constraints

1<\_ days <\_ 106

Example

Input

2

11101111

Output:

00000110

Explanation:

The street light at position 0 has its neighboring street lights 0(assumption)and 1.So, on the next day,it will be 1.

The street light at position 1 has both its neighboring street lights are 1. So,on the next day,it will be 0.

The street light at position 2 has one of its neighboring street,lights is 0 and the other one is 1. So, on next day,it will be 1.

The street light at position 3 is 0 and both its neighboring street lights are 1, So, on next day, the street light at position 3 will be 0 only.

Similarly,we can find the state of the remaining street lights for the next day.

So,the state of street lights after first day is 101010001

After two days,the state of street lights is 00000110

PROBLEM-1-3:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Mr. Woods, an electrician has made some faulty connections of eight street lights in Timberland city. The connections are such that if the street lights adjacent to particular light are both ON(represented as 1)or are both OFF (represented as 0),then that street light goes OFF the next night. Otherwise, it remains ON the next night. The two street lights at the end of the road have only a single adjacent street light, so the other adjacent light can be assumed to be always OFF. The state of the lights on a particular day is considered for the next day and not for the same day.

Due to this fault, the people of the city are facing difficulty in driving on the road at night. So, they have filed a complaint about this to the Head of the Federal Highway Administration. Based on this complaint the head has asked for the report of the state of street lights after M days.

Write an algorithm to output the state of the street lights after the given M days.

Input

The first line of the input consists of an integer-days, representing the number of days (M).

The next line consists of eight space-separated integers representing the current state of street lights i.e.eighter 0 or 1.

Output

Print eight space-separated integers representing the state of street lights after M days.

Constraints

1<\_ days <\_ 106

Example

Input

2

11101111

Output:

00000110

Explanation:

The street light at position 0 has its neighboring street lights 0 (assumption) and 1. So, on the next day, it will be 1.

The street light at position 1 has both its neighboring street lights are 1. So, on the next day, it will be 0.

The street light at position 2 has one of its neighboring street lights 0 and other one is 1. So, on next day, it will be 1.

The street light at position 3 is 0 and both its neighboring street lights are 1. So, on next day, the street light at position 3 will be 0 only.

Similarly, we can find the state of the remaining street lights for the next day.

So, the state of street lights after first day is 10101001

After two days, the state of street lights is 00000110

Code:

#include <stdio.h>

int main()

{

int n,i,a[100],b[50];

scanf("%d",&n);

for(i=0;i<8;i++){

scanf("%d",&a[i]);

}

for(i=1;i<=n;i++){

// a[-1]=0;

//a[n]=0;

if(a[1]==1){

b[0]=1;

}else{

b[0]=0;

}

if(a[i-1]==a[i+1])

b[i]=0;

else

b[i]=1;

}//not completed

TEST CASES: 1:

TestCase 1:

Input:

5135

73 4 63 33 65

Expected Return Value:

4 33

TestCase 2:

Input:

5 4 8

2 10 89 5 88

Expected Return Value

5

TEST CASES: 2:

TestCase1:

Input

1

10000100

Expected Return Value

01001010

TestCase2:

Input:

2

11101111

Expected Return Value:

00000110

TEST CASES: 3:

TestCase1:

Input

7

9-38-6-7 8 10

Expected Return Value:

3

TestCase2:

Input:

8

-20-16-9-3-2 8 9 22

Expected Return value:

5

TEST CASES: 4:

TestCase1:

Input:

1

10000100

Expected Return Value:

01001010

Testcase2:

Input:

2

11101111

Expected Return Value:

00000110

PROBLEM-2:1:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

The cloud computing company Cloud0 can accommodate various requirements for resources. The company system runs two servers. For load balancing purposes, the load of the resources gets transferred to the serves one by one. Initially , the first request goes to server 1, the next request goes to server 2, and so an. The requests served by the servers are two types i.e. one for memory deallocation (denoted by a negative number).

Write an algorithm to find the total number of units of memory allocated/deallocated by the server 1 after processing all the requests.

Input

The first line of the input consists of an integer numOfReq, representing the number of requests(N).

The second line consists of N space-separated integers-req1,req2,…,reqN representing the requests for the allocation/deallocation of the respective memory units.

Output

Return an integer representing the total number of units of memory allocation/deallocated by the server 1 after processing all the requests.

Constraints

0<\_ numOfReq<\_105

-106<\_reqi<\_106

0<\_i<numOfReq

Example

Input

7

2-3-8-6-7-1-81

Output

4

Explanation

The requests served by server 1 are [2,8,-7,1]

So, the total processing time of server 2 is 4.

TESTCASE-1:

TestCase1:

Input

5

14 53 2 23 1

Expected Return Value

17

TestCase2:

Input:

6

54 2 32 56 88 12

Expected Return Value:

174

Code:

#include <stdio.h>

int main()

{

int n,i,a[100],sum=0;

scanf("%d",&n);

for(i=0;i<n;i++){

scanf("%d",&a[i]);

}

for(i=0;i<n;i++){

if(i%2==0){

sum=sum+a[i];

}}

printf("%d",sum);

return 0;

}

PROBLEM-2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A pizza shop makes vegan pizzas as well as meat based pizzas. The customers place N orders at the shop and their order number gets printed on their bill. The shop displays k out of N both-vegan and meat based pizza orders on their display screen at a single time. The pizza shop is very famous and receives many orders. So to avoid confusion, the vegan pizza orders are displayed as a positive order number and the meat based pizza orders are displayed as a negative order number. All the orders are delivered in the order in which they are displayed on the screen. Each time a displayed order is ready, it is then removed from the display screen and the next order is added to the display at the end.

A couple has come to eat pizza with their child Billy . Billy is a very naughty child and to keep him busy, his parents tell him to makes a list of the first meat based pizza order number present in each set of k orders displayed on the shop’s display screen.

Write an algorithm to help Billy make a list of the first meat based pizza order numbers displayed on the screen each time an order is delivered to a customer.

**Input**

The first line of the input consists of two space-separated integers – numOfOrders and size, representing the total number of orders placed(N) and the number of orders displayed on the screen (K), respectively.

The second line consists of N space-separated integers representing the vegan pizza and meat based pizza order numbers of the orders placed by the customers.

**Output**

Print a list of space-separated integers representing the first meat based pizza order of every k orders displayed on the screen each time an order is delivered to a customer and print 0 if the screen does not display any meat based pizza order.

Constraints

0<\_ numOfOrders<\_106

0<\_size <\_ numOfOrders

-109<\_ ordersNum<\_109, where order Num represents the order numbers of the orders placed

Example

Input

63

-11-2 19 37 64-18

Output:

-11 -2 0-18

Explanation:

Step1: At the time of the first display, the order numbers displayed are [-11,-2,19].

So, the first displayed meat based pizza order number is -11.

Step2:At the time of the next display, the order numbers displayed are[-2,19,37]

So, the first displayed meat based pizza order number is -2.

Step3:At the time of the next display, the order numbers displayed are[19,37,64].

Since no meat based pizza order is displayed, the output is 0.

Step4:Similarly,at the time of the next display, the order numbers displayed are[37,64,-18].

So, the first displayed meat based pizza order number is -18.

So, the output is [-11, -2, 0,-18].

TESTCASE:

Testcase1:

Input:

63

-11 -2 19 37 64 -18

Expected Return Value:

-42 -42 -56 -56

PROBLEM-3:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Write an algorithm to find the number of occurrences of needle in given positive number haystack.

**Input**

The first line of the input consists of an integer needle, representing a digit.

The second line consists of an integer haystack, representing the positive number.

**Output**

Print an integer representing the number of occurrences of needle in haystack.

Constraints

0<\_needle<\_9

0<\_haystack<\_99999999

Example

Input:

2

123228

Output:

3

Explanation:

Needle 2 is occurring 3 times in the hastack.

TESTCASE:

TestCase1:

Input:

0

1025480

Expected Return Value:

2

TestCase2:

Input:

5

55555555

Expected Return Value:

8

Code:

**#include <stdio.h>**

**int main()**

**{**

**int n,rem,digit,c=0;**

**scanf("%d",&digit);**

**scanf("%d",&n);**

**while(n>0){**

**rem=n%10;**

**if(rem==digit)**

**c++;**

**n=n/10;**

**}**

**printf("%d",c);**

**return 0;**

**}**

PROBLEM-3A:1:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A company Digicomparts manufactures 52 types of unique products for laptop and desktop computers. It manufactures 10 types of laptop products and 42 types of desktop products.

Each product manufactured by the company has a unique product ID from a-z and A-Z. The laptop products have product IDs(a,i,e,o,u,A,I,E,O,U) while the rest of the product IDs are assigned to the desktop products. The company manager wishes to find the sales data for the desktop products.

Given a list of product IDs of the sales of the last N products. Write an algorithm to help the manager find the product IDs of the desktop products.

Input

The first line of the input consists of an integer numOfproducts, representing the number of products to be considered in the sales data(N).

The second line consists of N space-separated characters-prodID1,prodID2….,prodIDN representing the productIDs of the sales of the last N products.

Output

Print an integer representing the number of desktop products among the given sales data.

Constraints

0<\_numOfProducts<\_106

Example

Input

6

A v I k e l

Output

3

Explanation:

The productIDs of the desktop products in the sales data are [v,k,I].

So, the output is 3.

PROBLEM-3A:2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Martin is working as a commander in the army. The secret agent working for the army has shared a secret information with him. The secret information consists of a text and name of terrorist. With the given information, Martin needs to find the number of terrorists who are going to attack the army. He also gave you a hint that the number of terrorists is the number of times the terrorist’s name occurs in the text.

Write an algorithm to help Martin find the number of terrorists who are going to attack.

Input

The first line of the input consists of a string text, representing the text sent in the secret information shared by the secret agent.

The second line consists of a string name, representing the name of the terrorist.

Output

Print an integer representing the number of terrorists who are going to attack.

Note

The name matching is case insensitive.

Overlap can exist while searching the name of the terrorist in the text.

Example

Input

TimisplayinginthehouseofTimwiththetoysofTim

Tim

Output

3

Explanation:

Tim is occurring 3 times in the text of secret information.

Number of terrorists=3

TESTCASE:1:

Testase1:

Input:

6

A v I k e l

Expected Return Value:

3

TestCase2:

Input:

9

S d h a j m e k p

Expected Return Value:

7

PROBLEM-A:4:1:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A company has a sales record of N products for M days. The company wishes to know the maximum revenue received from a given product of the N products each day. Write an algorithm to find the highest revenue received each day.

Input

The first line of the input consists of two space-separated

Integers-days (M) and products(N), representing the days and the products in the sales record.

The next M lines consist of N space-separated integers representing the sales revenue received from each product each day.

Output

Print M space-separated integers representing the maximum received each day.

Example

Input

34

100 198 333 323

122 232 221 111

223 565 245 764

Output:

333 232 764

Explanation:

The maximum revenue received on the first day is 333, followed by a maximum revenue of 232 on the second day and a maximum revenue of 764 on the third day.

PROBLEM-A:4:2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Martin follows his father for a morning jog. His father starts at a position which is x1 meters away from their home and runs rectili nearly at a constant speed of v1 meters per step for N steps.

Martin is standing at x2 meters away from his home. He wonders as to how fast he must run at some constant speed of v2 meters per step so as to maximize F, where F equals the number of father’s footsteps visited by Martin during his run. It is given that the first step that Martin will land at from his starting position, will have to be visited by his father.

Note that if more than one prospective velocities of Martin result in the same number of maximum common steps, output the highest prospective velocity as v2.

Write an algorithm to help Martin calculate F and v2.

Input

The first line of the input consists of two space-separated integers.

-x1 and x2 representing the initial positions of the Martin’s father and Martin, respectively.

The second line consists of two space-separated integers

-v1 and N representing the velocity of the father and the number of steps taken by the father, respectively.

Output

Print two space-separated integers as maximum number of common footsteps F and respective speed v2.

Constraints

1<\_ x1<\_105

0<\_x2<\_x1

1<\_v1<\_104

1<\_N<\_104

Example

Input

32

220

Output:

211

Explanation:

Martin can have a maximum of 21 common footsteps with a velocity of 1m/step.

TESTCASE:1:

TestCase1:

Input:

44

721 2 8

8 18 16

16 12 9 7

12 0 7 2

Expected Return Value:

21 18 16 12

TestCase2:

Input:

3 3

98 -321 83

83 54 -75

283 46 68

Expected Return Value:

98 83 283

TESTCASE:2:

TestCase1:

Input:

1 0

1 2

Expected Return Value:

3 1

TestCase2:

Input:

13 5

9 4

Expected Return Value:

1 44

PROBLEM-A:5:1:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A company has a sales record of N products for M days. The company wishes to know the maximum revenue received from a given product of the N products each day. Write an algorithm to find the highest revenue received each day.

Input

The first line of the input consists of two space-separated

Integers-days(M) and products(N),representing the days and the products in the sales record.

The next M lines consist of N space-separated integers representing the sales revenue received from each product each day.

Output

Print M space-separated integers representing the maximum revenue received each day.

Example

Input

34

100 198 333 323

122 232 221 111

223 565 245 764

Output :

333 232 764

Explanation:

The maximum revenue received on the first day is 333, followed by a maximum revenue of 232 on the second day and a maximum revenue of 764 on the third day.

PROBLEM-A:5:2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Given a route in a straight line. N buses operate between various bus stations. There is workstation at the start of the route. The distance of the bus stations are calculated from the workstation. The transportation authority wants to decrease the number of buses that run in the city so if the routes of the buses overlap, then all such buses are replaced by a single bus. The authority wants to find the number of buses after the replacement of buses on the overlapping routes.

Write an algorithm to find the number of buses after the replacement of buses on the overlapping routes.

Input

The first line of the input consists of an integer-busCount, representing the number of buses running on the route(N).

The next N lines consist of two space-separated integers representing the distance of starting and ending bus stations of N buses from the workstation.

Output

Print an integer representing the number of buses running on the route after the replacement of buses on overlapping routes.

Constraints

0<\_busCount<\_1000

0<\_busStations[i][0]<busStations[i][1]<\_106,busStations are the starting and ending bus stations of a bus.

0<\_i<\_busCount

Example

Input

4

28

6 10

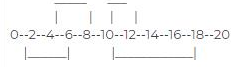
12 14

12 20

Output

2

Explanation:



The buses running between the bus stations [2,8] and [6,10] are combined to one.

Similarly, the buses running between the bus stations [12,14] and [12,20] are combined to one.

So, the total number of buses running on the route are 2.

TESTCASES:1:

TestCase1:

Input:

4 4

7 21 2 8

8 18 16

16 12 97

12 0 7 2

Expected Return Value:

21 18 16 12

Testase2:

Input:

3 3

98 -321 83

83 54 -75

283 46 68

Expected Return Value:

98 83 283

TESTCASES:2:

TestCase1:

Input:

4

2 8

6 10

12 14

12 20

Expected Return Value:

2

TestCase2:

6

10 17

34 42

56 65

100 101

105 110

115 119

Expected Return Value:

6

PROBLEM-A:6:1:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

You are playing an online game. In the game, a list of N numbers is given. The player has to arrange the numbers so that all the odd numbers of the list come after the even numbers. Write an algorithm to arrange the given list. Such that all the odd numbers of the list come after the even numbers.

Input

The first line of the input consists of an integer num, representing the size of the list (N).

The second line of the input consists of N space-separated integers representing the values of the list.

Output

Print N space-separated integers such that all the odd numbers of the list come after the even numbers.

Example

Input

8

10 98 12 22 3 33 21 11

Output:

10 98 12 22 3 33 21 11

Explanation:

All the even numbers are placed before all the odd numbers.

PROBLEM-A:6:2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

In a town, the houses are marked with English alphabets. A committee in the town wants to renovate houses in the town. As the funds are limited so they plan to renovate only the houses marked with vowels. The committee head gives the list of houses to the members and asks them to identify the houses which were not renovated.

Write an algorithm to help the committee members find the list of houses that were not renovated.

Input

The input consists of a string representing the sequence of house markings.

Output

Print a string representing the list of houses that were not renovated.

Constraints

All the house markings are of English alphabets.

Example

My name is Anthony

Output:

Mynmsnthny

Explanation:

The list of houses that were not renovated is Mynmsnthny.

TESTCASES:1:

TestCase1:

Input:

5

73 4 63 23 65

Expected Return Value:

4 73 63 23 65

TestCase2:

Input:

6

24 13 68 79 46 77

Expected Return Value:

24 68 46 13 79 77

TESTCASES:2:

TestCase1:

Input:

4 4

1 2 3 4

4 5 6 7

Expected Return value:

6

TestCase2:

Input:

4 7

4 3 5 6

3 2 1 7 9 8 13

Expected Return Value:

9

PROBLEM-A:7:1:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A company Selenia is planning a big sale at which they will give their customers a special promotional discount. Each customer that purchases a product from the company has a unique customer ID numbered from 0 to N-1. Andy, the marketing head of the company, has selected bill amounts of the N customers for the promotional scheme. The discount will be given to the customers whose bill amounts are perfect squares. The customers may use this discount on a future purchase.

Write an algorithm to help Andy find the number of customers that will be given discounts.

Input

The first line of the input consists of an integer numOfCust, representing the number of customers whose bills are selected for the promotional discount(N).

The second line consists of N space-separated integers-bill1, bill2….. ,billN representing the bill amounts of the N customers selected for the promotional discount.

Output

Print an integer representing the number of customers that will be given discounts.

Constraints

0<\_numOfCust<\_106

0<\_billi<\_106

0<\_i<numOfCust

Example

Input

6

25 77 54 81 48 34

Output:

2

Explanation:

The bill amounts that are perfect squares are 25 and 81.

So, the output is 2.

PROBLEM-A:7:2:

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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

You are given two lists of different lengths of positive integers. Write an algorithm to count the number of elements which are not common in the given lists.

Input

The first line of the input consists of two space-separated integers-length1 and length2, an integer representing the number of elements in the first list and the number of elements in the second list, respectively.

The second line consists of an integer representing the first list of positive integers.

The third line consists of an integer representing the second list of positive integers.

Output

Print a positive integer representing the count of elements which are not common in both the given lists.

Example

Input

11 10

1 1 2 3 4 5 5 7 6 9 10

11 12 13 4 5 6 7 18 19 20

Output:

12

Explanation:

The numbers which are not common in both the lists are [1,1,2,3,9,10,11,12,13,18,19,20].

So, the output is 12.

TESTCASES:1:

TestCase1:

Input:

4

3 5 7 16

Expected Return Value:

1

TestCase2:

Input:

6

25 77 54 81 48 34

Expected Return Value:

2

TESTCASES:2:

TestCase1:

Input:

4 4

1 2 3 4

4 5 6 7

Expected Return value:

6

TestCase2:

Input:

4 7

4 3 5 6

3 2 1 7 9 8 13

Expected Return Value:

9

9-2.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.

**Mr. Jason has captured your friend and has put a collar around his neck. He has locked it with a given ‘locking keg’ it can only be opened now with an ‘unlocked key’ Your friend has seen the ‘locking key’ he does not know about the ‘unlocking key’. Given the locking key, one can figure out the ‘unlocking key’ which is the smallest (in magnitude) permutation of the digits of that number and it never starts with zero.**

**Help your friend to write an algorithm that takes the locking key as an input and outputs the unlocking key.**

**Input**

**The input consists of an integer K, representing the locking key.**

**Output**

**Print an integer representing the unlocking key.**

**Constraints**

**-107 ≤ *k* ≤ 107**

**Note**

**TEST CASES**

Test Cases1:

**Input:**

310

**Expected Return Value:**

103

**Test Cases2:**

**Input:**

918

**Expected Return Value:**

189

1).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 is planning to give a special discount on all its product to its product to its customers for the Christmas 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 affer 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 con 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 two space-separated integers – *numOfProducts* and *disAmount*, representing the number of different types of products (N) and the discount amount that will be distributed among the customers.

Order N representing the current status of the stock for the orders of the respective product types.

**Output**

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

**Constraints**

O ≤ *numOfProducts, disAmount* ≤105

-106 ≤ *order i* ≤106

0≤ i ≤ *numOfProducts*

**Example**

**Input:**

7 18

9 – 13 8 – 7 – 8 18 10

**Output**

2

**Explanation:**

The product for which the number of customers will collect the discount amount “3” are for product types 0 and 5, i.e. 9 and 18, respectively.

Sao, the output is 2

**Test Case 1:**

Input:

6 3 6

2 3 5 6 8 11

**Expected Return Value:**

3

**Test Case 2:**

Input:

6 18

6 9 1 3 2 18

**Expected Return Value:**

6

2).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 alternate sort of a list consists of a list consists of alternate elements (starting from the first position) of the given list after sorting it in an ascending order. You are given are given a list of unsorted elements. Write an algorithm to find the alternate sort of the given list.

Input

The first line of the input consists of an integer *size*, representing the size of the given list (N).

The second line consists of N space-separated integers – *list[1], list[2]. ….. , list[N],* representing the input list of integers.

Output

Print space-separated integers representing alternate sorted elements of the given list.

Constraints

0 <size ≤ 106

-106 ≤ *list[i]* ≤106

0 ≤ i < size

Example

Input:

8

3 5 1 5 9 10 2 6

Output

1 3 5 9

Explanation:

After sorting, the list is [1, 2, 3, 5, 5, 6, 9, 10]

So , the alternate elements of the sorted list are [1, 3, 5, 9]

TEST CASES

Test Cases 1:

Input

8

3 5 1 5 9 10 2 6

Expected Return Value:

1 3 5 9

Test Cases 2:

Input

9

16 15 11 8 5 4 3 2 1

Expected Return Value:

1 3 5 11 16

3). 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.

Ethan is the team leader of a team of N members. Ethan has assigned an error score to each member in his team based on the bugs that he has found in that particular team member’s task. Since, the error score has increased to a significant large value, he wants to give all the team members a chance to make it zero and improve their reputation in the organization. To incorporate this he introduces a new rule that whenever a team member completes a project successfully, the error score of that member decreases by a count P and the error score of all the other team members whose score is greater than zero decreases by a count Q.

Write an algorithm to help Ethan find the minimum number of projects that the team required to complete to make the error score of all the team members as zero.

Input

The first line of the input consists of three space-separated integers – N, P and Q representing the total number of team members, the count by which the error score of the team member who completed a project successfully decreases and the count by which the error score of the team member whose error score is greater than zero decreases, respectively.

The second line consists of N space-separated integers resenting the initial error scores of the team members.

Output

Print an integer representing the minimum number of projects that the team required to complete to make the error score of all the team members as zero.

Constraints

1 ≤ N ≤ 2\*10 5

1 ≤ Q ≤ P ≤10 9

*0 ≤ Mi ≤ 109; M*i represents the error score of the ith team member

*1 ≤ I ≤ N*

Note

The error score of any team member can never be less than zero.

Example

Input:

3 4 1

6 4 1

Output

3

Explanation:

Once the first team member completes a project, the updated error score of the team members is: 2 3 0

After the second member completes the project, the updated error score of the team members is : 1 0 0

Again, after the first member completes the project, the updated error score of the team members is : 0 0 0

So, the team needs to complete 3 projects to make the error score of all the team members as zero

TEST CASES

Test Cases 1:

Input

4 3 1

9 8 2 5

Expected Return Value:

5

Test Cases 2:

Input

3 2 1

4 3 2

Expected Return Value:

3

4). 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 Selenia is planning a big sale at which they will give their customers a special promotional discount. Each customer that purchases a product from the company has a unique customerlD numbered from 0 to N-1. Andy, the marketing head of the company, has selected bill amounts of the N customers for the promotional scheme. The discount will be given to the customers whose bill amounts are perfect squares. The customers may use this discount on a future purchase.

Write an algorithm to help Andy find the number of customers that will be given discounts.

Input

The first line of the input consists of an integer *numOfCust*, representing the number of customers whose bills are selected for the promotional discount (N).

The second line consists of N space-separated integers – bill1, bill2 ….. bill representing the bill amounts of the N customers selected for the >>>>>>>>>

TEST CASES

Test Cases 1:

Input

4

3 5 7 16

Expected Return Value:

1

Test Cases 2:

Input

6

25 77 54 81 48 34

Expected Return Value:

2

1).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.

Andrew is a stock trader who trades in N selected stocks. He has calculated the relative stock price changes in the N stocks from the previous day stock prices. Now, his lucky number is K, so he wishes to invest in the particular stock that has Kth smallest relative stock value.

Write an algorithm for Andrew to find the Kth smallest stock price out of the selected N stocks.

**Input**

The first line of the input consists of two space-separated integers – *numOfStocks and valuek,* representing the number of selected stocks (N) And the value K for which he wishes to find the stock price, respectively.

The second line consists of N space-separated integers – stock1, stock2, ……, stock N representing the relative stock prices of the selected stocks.

**Output**

Print an integer representing the Kth smallest stock price of selected N stocks.

**Constraints**

0 < *valueK*  ≤ *numOfStocks* ≤ 106

0 ≤ *stocki* ≤ 106

0 ≤ i < *numOfStocks*

**Example**

**Input:**

7 5

9 -3 8 -6 -7 18 10

**Output**

9

**Explanation:**

The sorted relative stock prices are [-7, -6, -3, 8, 9, 10, 18]

So, the 5th smallest stock price is 9.

**TEST CASES**

Test Cases1:

**Input:**

7 4

1 2 3 4 5 7 9

**Expected Return Value:**

4

**Test Cases2:**

**Input:**

5 3

10 5 7 88 19

**Expected Return Value:**

10

2). 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.

Write an algorithm to find the number of occurrences of needle in given positive number haystack.

**Input**

The first line of the input consists of an integer needle, representing a digit. The second line consists of an integer haystack, representing the positive number.

**Output**

Print an integer representing the number of occurrences of needle in haystack.

**Constraints**

0 ≤ needle ≤ 9

0 ≤ haystack ≤ 99999999

**Example**

**Input:**

2 123228

**Output**

3

**Explanation:**

*Needle* 2 is occurring 3 times in the *hastack.*

**TEST CASES**

**Test Cases1:**

**Input:**

0

1025480

**Expected Return Value:**

2

**Test Cases2:**

**Input:**

5

55555555

**Expected Return Value:**

10

3).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.

The cloud computing company CloudO can accommodate various requirements for resources. The company system runs two servers. For load balancing purposes, the load of the resources gets transferred to the servers one by one. Initially, the first request goes to server 1, the next request goes to server 2, and so on. The requests served by the servers are of two types i.e. one for memory allocation (denoted by a positive number) and the other for memory deallocation (denoted by a negative number).

Write an algorithm to find the total number of units of memory allocated/deallocated by the server 1 after processing all the requests.

**Input**

The first line of the input consists of an integer *numOfReq,* representing the number of requests ( N)

The second line consists of N space-separated integers – req1, req2, ……. , reqN representing the requests for the allocation/ deallocation of the respective memory unoits.

**Output**

Return an integer representing the total number of units of memory allocated/deallocated by the server 1 after processing all the requests.

**Constraints**

O ≤ *numOfReq* ≤105

-106 ≤ *req i* ≤106

0 ≤ i < *numOfReq*

**Example**

**Input:**

7

2 -3 8 -6 -7 18 1

**Output**

4

**Explanation:**

The requests served by server 1 are [2,8,-7,1].

So, the total processing time of server 2 is 4.

**TEST CASES**

**Test Cases1:**

**Input:**

5

14 53 2 23 1

**Expected Return Value:**

17

**Test Cases2:**

**Input:**

6

54 2 32 56 88 12

**Expected Return Value:**

174

4.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 employee in an organization has started working on N projects (numbered 1 to N). Every week he/she can work on a module of one of the projects. The modules that are chosen on any two successive weeks should be of a different project and a project I can have at most Ci modules. The modules of the projects are such that a module is completed in a week.

Write an algorithm to determine the number of weeks for which he/she can work on projects following the above-mentioned rules.

**Input**

The first line of the input consists of an integer N, representing the number of projects.

The next line consists of N space-separated integers – C1, C2, .., CN representing the number of modules of the projects.

**Output**

Print an integer representing the maximum number of weeks for which an employee can work on the projects.

**Constraints**

*1 ≤ N≤ 5\*104*

*1 ≤ Ci ≤ N*

*∑ Ci ≤ 105*

*1 ≤ I ≤ N*

**Example**

**Input**

3

7 2 3

**Output**

11

**Explanation:**

The first, second and third project have 7, 2 and 3 modules respectively. Themodules of different projects are selected on successive days in a sequence: first, second, first, third, first, second, first , third, first, third, first.

So, the maximum number of weeks an employee can work on these projects is 11.

**TEST CASES**

**Test Cases1:**

**Input:**

4

2 2 2 2

**Expected Return Value:**

8

**Test Cases2:**

**Input:**

3

7 2 3

**Expected Return Value:**

11

12-2). 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 pizza shop makes vegan pizzas as well as meat based pizzas. The customers place N orders at the shop and their order number gets printed on their bill. The shop displays K out of N both – vegan and meat based pizza orders on their display screen at a single time. The pizza shop is very famous and receives many orders. So to avoid confusion, the vegan pizza orders are displayed as a positive order number and the meat based pizza orders are displayed as a negative order number. All the orders are delivered in the order in which they are displayed on the screen. Each time a displayed order is ready, it is then removed from the display screen and the next order is added to the display at the end.

A couple has come to eat pizza with their child Billy. Billy is a very naughty child and to keep him busy, his parents tell him to make a list of the first meat based pizza order number present in each set of K orders displayed on the shop’s display screen.

Write an algorithm to help Billy make list of the first meat based pizza order numbers displayed on the screen each time an order is delivered to a customer.

**Input**

The first line of the input consists of two space-separated integers – *numOfOrders* and size, representing the screen (K), respectively.

The second line consists of N space-separated integers representing the vegan pizza and meat based pizza order numbers of the orders placed by the customers.

**Output**

Print a list of space-separated integers representing the first meat based pizza order of every K orders displayed on the screen each time an order is delivered to a customer and print 0 if the screen does not display any meat based pizza order.

**Constraints**

0 ≤ numOfOrders ≤ 106

0 ≤ size ≤ numOfOrders

-109 ≤ ordersNum≤ 109 ; where orderNum represents the order numbers of the orders placed

**Example**

Input:

63

-11 -2 19 37 64 -18

Output:

-11 -2 0 -18

**Explanation:**

Step1: At the time of the first display, the order numbers displayed are [-11,-2,19].

So, the first displayed meat based pizza order number is -11.

Step2:At the time of the next display, the order numbers displayed are[-2,19,37]

So, the first displayed meat based pizza order number is -2.

Step3:At the time of the next display, the order numbers displayed are[19,37,64].

Since no meat based pizza order is displayed, the output is 0.

Step4:Similarly,at the time of the next display, the order numbers displayed are[37,64,-18].

So, the first displayed meat based pizza order number is -18.

So, the output is [-11, -2, 0,-18].

**TEST CASES**

Test Cases1:

**Input:**

6 3

-11 -2 19 37 64 -18

**Expected Return Value:**

-11 -2 0 -18

**Test Cases2:**

**Input:**

7 4

35 -42 23 -56 -84 92 39

**Expected Return Value:**

-42 -42 -56 -56

**1). Repeat**

1) 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 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.

**Output:**

Print space-separated strings 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.

**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”].

**TEST CASES**

Test Cases1:

**Input:**

Block brick black blue blah black

**Expected Return Value:**

black

**Test Cases2:**

**Input:**

Cat batman latt matter cat matter cat

**Expected Return Value:**

Cat matter

2). 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.

In a town, the houses marked with English alphabets. A committee in the town wants to renovate houses in the town. As the funds are limited so they plan to renovate only the houses marked with vowels. The committee head gives the list of houses to the members and asks them to identify the houses which were not renovated.

Write an algorithm to help the committee members find the list of houses that were not renovated.

**Input:**

The input consists of a string representing the sequence of house markings.

**Output:**

Print a string representing the list of houses that were not renovated.

**Constraints:**

All the house markings are of English alphabets.

**Example:**

MynameisAnthony

Output:

Mynmsnthny

**Explanation:**

The list of houses that were not renovated is Mynmsnthny

**TEST CASES**

Test Cases1:

**Input:**

bacdefghijkmnopqrstu

**Expected Return Value:**

bedfghjklmnpqrst

**Test Cases2:**

**Input:**

bacdefgh

**Expected Return Value:**

bcdfgh

1).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 Digicomparts manufactures 52 types of unique products for laptop and desktop computers. It manufactures 10 types of laptop products and 42 types of desktop products.

Each product manufactured by the company has a unique product ID from a-z and A-Z. The laptop products have product IDs (a,i,e,o,u,A,I,E,O,U) while the rest of the product IDs are assigned to the desktop products. The company manager wishes to find the sales data for the desktop products.

Given a list of product IDs of the sales of the last N products, write an algorithm to help the manager find the product IDs of the desktop products.

Input

The fist line of the input consists of an integer *numOfProducts*, representing the number of products to be considered in the sales data (N).

The second line consists of N space-separated characters – *prod ID1, prodID2, …., prod IDN* representing the product IDs of the sales of the last N products.

Output

Print an integer representing the number of desktop products among the given sales data.

Constraints

0 ≤ *numOFProducts* ≤ 106

Example

Input:

a v I k e l

output

3

Explanation:

The product IDs of the desktop products in the sales data are [v, k, l].

So, the output is 3

**TEST CASE**

**Test Case 1:**

Input:

6

a v I k e l

**Expected Return Value:**

3

**Test Case 2:**

Input:

9

s d h a j m e k p

**Expected Return Value:**

7

2). 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.

You are given a list of integers and an integer K. Write an algorithm to find the number of elements in the list which are strictly less than K.

Input

The first line of the input consists of two space-separated integers – size and num K, representing the number of elements in the list (N) and the integer to be compared (K).

The second line consists of N space- separated integers- *elements[0], elements[1], ……. Elements[N-1]* representing the list of integers.

Output

Print an integer representing the number of elements in the list which are strictly less than *num.*

Constraints

-109 ≤*numK* < 109

-109≤ *elements[0], elements[1], …………elements[N-1]* 109

Example

Input:

7 5

1 7 4 5 6 3 2

Output:

4

Explanation:

The numbers which are less than 5 in elements are 1, 2, 3, 4, .

So, the output is 4

**TEST CASE**

**Test Case 1:**

Input:

8 -5

11 25 66 -4 77 55 -10

**Expected Return Value:**

0

**Test Case 2:**

Input:

6 2828

243 516 1846 -488 791 -430

**Expected Return Value:**

6

15-4). 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.

You are given a list of integers of size N. Write an algorithm to sort the first keleaments (from list [O] to list[K-1])of this list in an ascending order and the remaining (list[K] to list [N-1]) elements in a descending order.

Input

The first line of the input consists of two space-separated integers – len and num representing the number of elements in the list (N) and the integer of elements to be sorted in an ascending order (K).

The second line consists of N space- separated integers representing the elements of the list.

Output

Print a list of integer representing the required sorted list.

Constraints

Num < len

Example

Input:

8 3

11 7 5 10 46 23 16 8

Output:

5 7 11 46 23 16 10 8

Explanation:

We have to arrange the first three elements in an increasing order and the remaining elements in the decreasing order.

The final list = [5, 7, 11, 46, 23, 16, 10, 8]

**TEST CASE**

**Test Case 1:**

Input:

8 4

7 43 12 4 1 3 78 6

**Expected Return Value:**

4 7 12 43 78 6 3 1

**Test Case 2:**

Input:

10 5

1 2 3 4 5 6 7 8 9 10

**Expected Return Value:**

1 2 3 4 5 10 9 8 7 6

15-2). 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.

You are given two lists of different lengths of positive integers. Write an algorithm to count the number of elements which are not common in the given list.

**Input**

The first line of the input consists of two space-separated integers – length 1 and length2, an integer representing the number of elements in the first list and the number of elements in the second list, respectively.

The second line consists of an integer representing the first list of positive integers.

The third line consists of an integer representing the second list of positive integers.

**Output**

Print a positive integer representing the count of elements which are not common in both the given lists.

**Example**

Input:

11 10

1 1 2 3 4 5 5 7 6 9 10

11 12 13 4 5 6 7 18 19 20

Output:

12

Explanation:

The numbers which are not common in both the lists are [1,1 2, 3, 9, 10, 11, 12, 13, 18, 19, 20]

So, the output is 12.

15-1

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 cold storage company has N storage units for various products. The company has received N orders that must be preserved at respective N temperatures in side the storage units. The company manager wishes to identify which products must be preserved at negative temperatures.

Write an algorithm to help the manager find number of products that have negative temperature storage requirements.

Input

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

The second line consists of N space-separated integers – *temp1, temp2,… , tempN* representing the temperatures at which the products must be preserved in the storage units.

Output

Print an integer representing the number of products that must be preserved at negative temperatures.

Constraints

O ≤ *numOfProducts* ≤ 106

106 ≤ tempi ≤ 106

*O ≤ i < numOfProducts*,

Example

Input

7 9 -3 8 -6 -7 8 10

Output

3

Explanation:

The products that must be preserved at negative temperatures are at indices [1,3,4] ie. [-3 -6 -7]

So, the output is

16-1The 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.

You are playing an online game. In the game, a list of N numbers is given. The player has to arrange the numbers so that all the odd numbers of the list come after the even numbers. Write an algorithm to arrange the given list such that all the odd numbers of the list come after the even numbers.

Input

The first line of the input consists of an integer num, representing the size of the list (N).

The second line of the input consists of N space-separated integers representing the values of the list.

Output

Print N space-separated integers such that all the odd numbers of the list come after the even numbers.

Example

Input

8

10 98 3 33 12 22 21 11

Output:

10 98 12 22 3 33 21 11

Explanation:

All the even numbers are placed before all the odd numbers.

**TEST CASES**

Test Cases1:

**Input:**

5

73 4 63 23 65

**Expected Return Value:**

4 73 63 23 65

**Test Cases2:**

**Input:**

6

24 13 68 79 46 77

**Expected Return Value:**

24 68 46 13 79 77

16-2. 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 alternate sort of a list consists of a alternate elements (starting from the first position) of the given list after sorting it in an ascending order. You are given a list of unsorted elements. Write an algorithm to find the alternate sort of the given list.

Input

The first line of the input consists of an integer size, representing the size of the

Given list (N).

The second line consists of N space-separated integers - ; list[1], list[2], …., list[N], representing the input list of integers.

Output

Print spa-separated integers representing alternate sorted elements of the given list.

Constraints

O < size ≤ 106

-106 ≤ list[i] ≤ 106

O ≤ I < size

Example

input

8

3 5 1 5 9 10 2 6

Output

1 3 5 9

Explanation:

After sorting, the list is [1, 2, 3, 5, 5, 6, 9, 10]

So, the alternate elements of the sorted list are [1, 3, 5, 9]

**TEST CASES**

Test Cases1:

**Input:**

8

3 5 1 5 9 10 2 6

**Expected Return Value:**

1 3 5 9

**Test Cases2:**

**Input:**

9

16 15 11 8 7 4 3 2 1

**Expected Return Value:**

1 3 5 11 16

17-2The 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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

Martin is working as a commander in the army. The secret agent working for the army has shared a secret information with him. The secret information consists of a text and name of terrorist. With the given information, Martin needs to find the number of terrorists who are going to attack the army. He also gave you a hint that the number of terrorists is the number of times the terrorist’s name occurs in the text.

Write an algorithm to help Martin find the number of terrorists who are going to attack.

**Input**

The first line of the input consists of a string text, representing the text sent in the secret information shared by the secret agent.

The second line consists of a string name, representing the name of the terrorist.

Output

Print an integer representing the number of terrorists who are going to attack.

**Note**

The name matching is case insensitive.

Overlap can exist while searching the name of the terrorist in the text.

**Example**

Input

TimisplayinginthehouseofTimwiththetoysofTim

Tim

Output

3

**Explanation:**

Tim is occurring 3 times in the text of secret information.

Number of terrorists=3

**TESTCASE:1:**

**Testase1:**

Input:

Abcdefg

bcd

Expected Return Value:

1

TestCase2:

**Input:**

TimisplayinginthehouseofTimwiththetoysoflim

Tim

**Expected Return Value:**

3

17-1 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 the case of syntax/runtime error. The version of GCC being used is 5.5.0.

A company has a sales record of N products for M days. The company wishes to know the maximum revenue received from a given product of the N products each day Write an algorithm to find the highest revenue received each day.

**Input**

The first line of the input consists of two space-separated integers- days (M) and products (N), representing the days and the products in the sales record.

The next M lines consist of N space- separated integers representing the sales revenue received from each product each day.

**Output**

Print M space-separated integers representing the maximum revenue received each day.

**Example**

**Input**

3 4

100 198 333 323

122 232 221 111

223 565 245 754

**Output**

333 232 764

**Explanation:**

The maximum revenue received on the first day is 333, followed by a maximum revenue of 232 on the second bay and a maximum revenue of 764 on the third day

**TESTCASE:1:**

**Testase1:**

**Input:**

4 4

7 21 2 8

8 18 1 6

16 12 9 7

12 0 7 2

**Expected Return Value:**

21 18 16 12

**TestCase2:**

Input:

3 3

98 -321 83

83 54 -75

283 46 68

Expected Return Value:

98 83 283