Wipro NLTH Previous Year All Slot Programming Answers [11th Jan 2021 to 17th Jan 2021]

1) Security Key

Description

A company is transmitting data to another server. The data is in the form of numbers. To secure the data during transmission, they plan to obtain a security key that will be sent along with the data. The security key is identified as the count of the repeating digits in the data.

Write an algorithm to find the security key for the data.

Input

The input consists of an integer data, representing the data to be transmitted.

Output

Print an integer representing the security key for the given data. If no data is repeated it should display

-1

Constraints

NA

Example

Input:

578378923

Output: 3

Explanation

The repeated digits in the data are 7, 8 and 3. So, the security key is 3.

Public Test Cases

#	Input	Expected Output
1	578378923	3

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    long int n;
    scanf("%ld", &n);
    int a[10]={0};
    while(n)
    {
        a[n%10]++;
        n=n/10;
    }
    int i,sk=0;
    for(i=0;i<10;i++)
    {
        if(a[i]>1)
            sk++;
    }
    if(sk==0)
        sk=-1;
    printf("%d",sk);
}
```

2)

Encode as Number

Description

A company wishes to encodes its data. The data is in the form of a number. They wish to encode the data with respect to a specific digit. They wish to count the number of times the specific digit reoccurs in the given data so that they can encode the data accordingly. Write an algorithm to find the count of the specific digit in the given data.

Input

The input consists of two space-separated integers- data and digit, representing the data to be encoded and the digit to be counted in the data.

Output

Print an integer representing the count of the specific digit.

Constraints

NA

Example

Input:

572378233 3

Output:

3

Explanation

The digit 3 is repeated three times in the data. So the output is 3.

Public Test Cases

#	Input	Expected Output
1	572378233 3	3

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    long int n;
    scanf("%ld", &n);
    int k;
    scanf("%d", &k);
    int  c=0;
    while(n)
    {
        if(k==n%10)
            c++;
        n=n/10;
    }
    printf("%d",c);
}
```

3)

Sum of Adjacent Distances

Difficulty: easy
Max Points: 10

Description

Write a program to calculate and return the sum of distances between the adjacent numbers In an array of positive integers.

Note:

 You are expected to write code in the find Total distance function only which receive the first parameter as the number of items in the array, and second parameter as the array itself. You are not requested to take input from the console.

Constraints

Example

Finding the total distance between adjacent items of a list of 5 numbers

Input:

5

10 11 7 12 14

Output:

12

Explanation

The first parameters (5) is the size of the array next is an array of integers the total of distance is 12 as per the calculation below.

10-11=1

11-7=4

7-12=5

12-14=2

Total distance-1+4+5+2=12

Public Test Cases

#	Input	Expected Output
1	5 10 11 7 12 14	12

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    int i,a[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    int sum=0,t;

    for(i=0;i<n-1;i++)
    {
        t=a[i]-a[i+1];
        if(t<0)
            t=-t;
```

```
          sum=sum+t;
    }
    printf("%d",sum);
}
```

4)

Odd Even Online Game

Difficulty: easy
Max Points: 10

Description

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.

Constraints

NA

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.

Public Test Cases

#	Input	Expected Output
1	8 10 98 3 33 12 22 21 11	10 98 12 22 3 33 21 11

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    int i,a[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);

    }
    for(i=0;i<n;i++)
    {
        if(a[i]%2==0)
            printf("%d ",a[i]);
    }
    for(i=0;i<n;i++)
    {
        if(a[i]%2!=0)
            printf("%d ",a[i]);
    }
}
```

5)

One Time Password

Difficulty: easy
Max Points: 10

Description

An e-commerce site wishes to enhance its ordering process. They plan to implement a new scheme of OTP (One Time Password) generation for order confirmations. The OTP can be any number of digits. For OTP generation, the user will be asked to enter two random numbers. The first number entered should always be smaller than the second number. The OTP is calculated as the sum of the maximum and the minimum prime values in the range of the user-entered numbers.

Write an algorithm to find the OTP

Input

 The input consists of two space-separated integers – firstNumber and secondNumber, representing the two numbers entered by the user. Both numbers are considered in the range.

Output

Print an integer representing the sum of the largest and smallest prime numbers in the range of given numbers.

Constraints

-109< firstNumber <secondNumbers<109

Example

Input:

-97 50

Output:

50

Explanation

The smallest and largest prime numbers within the given numbers are -97 and 47, respectively. The sum of -97 and 47 is 50. So, the output is 50.

Public Test Cases

#	Input	Expected Output
1	-97 50	50

Programming Language: C (GCC 9.2.0)

```c
#include<stdio.h>
int main()
{
    long long int x,y,t;
    scanf("%lld %lld",&x,&y);
    if(x>y)
    {
        t=x;
        x=y;
        y=t;
    }
    long long int i,maxp,minp;
    while(1)
    {
        if(x<0)
            t=-x;
        else
            t=x;
        for(i=2;i<=t/2;i++)
        {
            if(t%i==0)
```

```
                break;
        }
        if(i>t/2)
        {
                minp=x;
                break;
        }
```

```
        x++;
    }
    while(1)
    {
        if(y<0)
            t=-y;
        else
            t=y;
        for(i=2;i<=t/2;i++)
        {
                if(t%i==0)
                break;
        }
        if(i>t/2)
        {
                maxp=y;
                break;
        }
        y--;
    }
    printf("%lld", maxp+minp);
    return 0;
}
```

6)

SecretMessage agency

Difficulty: easy
Max Points: 10

Description

SecretMessage agency provides message encoding and decoding services for secure data transfer. The first step in decoding includes removal of special characters and the whitespaces from the message, as special characters and whitespaces do not hold any meaning.

Write an algorithm to help the agency find the number of special characters and whitespaces in a given message.

Input

The input consists of a string message, representing the message that need to be decoded by the agency.

Output

Print an integer representing the number of special characters and whitespaces present in a given message.

Constraints

NA

Example

Input:

gasgg54@#vscsd!s*

Output:

4

Explanation

The special characters having no meaning are ('@','#';'!','*'].

Public Test Cases

#	Input	Expected Output
1	gasgg54@#vscsd!s*	4

Programming Language: C (GCC 9.2.0)

```c
#include<stdio.h>
int main()
{
    char text[1000];
    scanf("%[^\n]s",text);
    int i=0,c=0;
    while(text[i]!='\0')
    {
        if(!(
            (text[i]>='a' && text[i]<='z') ||
            (text[i]>='A' && text[i]<='Z') ||
            (text[i]>='0' && text[i]<='9') ) )
            c++;
        i++;
    }
    printf("%d",c);

}
```
7)

Garments company Apparel

Difficulty: easy
Max Points: 10

Description

The garments company Apparel wishes to open outlets at various locations. The company shortlisted several plots in these locations and wishes to select only plots that are square-shaped.

Write an algorithm to help Apparel find the number of plots that it can select for its outlets.

Input

The first line of the input consists of an integer num0fMots, representing the number of plots shortlisted by the company for outlets (N).

The second line consists of N space-separated integers - areal, areal, ..., areaN representing the area of the N plots selected for outlets.

Output

Print an integer representing the number of plots that will be selected for outlets.

Constraints

$0 < num0fPlots < 106$

$0 < area < 106$

$0 < i < num0fPlots$

Example

Input:

8

79 77 54 81 48 34 25 16

Output:

3

Explanation

The areas that are in square form are 81, 25 and 16. So, the output is 3.

Public Test Cases

#	Input	Expected Output
1	8 79 77 54 81 48 34 25 16	3

Programming Language: C (GCC 9.2.0)

```c
#include<stdio.h>
int main()
{
    int n;
    scanf("%d",&n);
    int a[n],i;
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);

    }
    int sroot,c=0;
    for(i=0;i<n;i++)
    {
        for(sroot=2;sroot*sroot<a[i];sroot++)
            ;
        if(sroot*sroot==a[i])
            c++;

    }
    printf("%d",c);
}
```

8)

Bucketize Ids

Difficulty: easy
Max Points: 10

Description

A company wishes to bucketize their item IDs for better search operations. The bucket for the item ID is chosen on the basis of the maximum value of the digit in the item ID. Write an algorithm to find the bucket to which the item ID will be assigned.

Input

The input consists of an integer item1D, representing the identity number of the item.

Output

 Print an integer representing the bucket to which the item ID will be assigned.

Constraints

NA

Example

Input:

32387634

Output:

8

Explanation

8 is the maximum digit value in the given item ID. So, the output is 8.

Public Test Cases

#	Input	Expected Output
1	32387634	8

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    long int n;
    scanf("%ld",&n);
    int m;
    m=n%10;
    while(n)
    {
        if(m>n%10)
            m=n%10;
        n=n/10;
    }
    printf("%d",m);
    return 0;
}
```

9)

Bingo-Online Lottery Game

Difficulty: easy
Max Points: 10

Description

A game company has designed an online lottery game, Bingo. In this game, N number cards are displayed. Each card has a value on it. The value can be negative or positive. The player must

choose two cards. To win the game, the product of the values of the two cards must be the maximum value possible for any pair of cards in the display. The winning amount will be the sum of the two cards chosen by the player.

Write an algorithm to find the winning amount as the sum of the values of the two cards whose product value is maximum.

Input

The first line of the input consists of an integer numCards, representing the number of cards (N). The second line consists of N space-separated integers - ..... , val1 , val2…., valN representing the values on the cards.

Output

Print an integer representing the sum of the values of the two cards whose product value is maximum.

Constraints

0 <numCardss106

-106< vali< 106

 0 < i < numCards

Example

Input:

7 9 -3 8 -6 -7 8 10

Output:

19

Explanation

 The maximum product of the values is 90, i.e. 9*10.

So, the sum of the values of the selected cards is 19.

Public Test Cases

#	Input	Expected Output
1	7 9 -3 8 -6 -7 8 10	19

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
```

```
{
    int n;
    scanf("%d",&n);
    int a[n],i,j,prod[n-1],t;
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-1-i;j++)
        {
            if(a[j]>a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
        }

    }
    for(i=0;i<n-1;i++)
    {
        prod[i]=a[i]*a[i+1];
    }
    int sum;
    if(prod[0]>prod[n-2])
        sum=a[0]+a[1];
    else
        sum=a[n-1]+a[n-2];
    printf("%d",sum);
    return 0;
}
```

10)

Digital Secure Data Transfer Solutions

Difficulty: easy
Max Points: 10

Description

The company Digital Secure Data Transfer Solutions provides data encryption and data sharing services. Their process uses a key K for encryption when transmitting characters. To encrypt a character, the character is replaced by the following Kth character with the same case in the English alphabet set. The English alphabetical set is considered in a cyclic fashion for the last K characters.

Write an algorithm to find the character used to encrypt the given character.

Input

The input consists of a space-separated character and integer - character and key, representing the character to be transmitted, and the key (K), respectively.

Output

Print a character representing the encrypted character.

Constraints

NA

Example

Input: D 3

Output: G

Explanation

Replace D with G.

So, the output is G.

Public Test Cases

#	Input	Expected Output
1	D 3	G

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    char ch;
    int k;
    scanf("%c",&ch);
    scanf("%d",&k);
    k=k%26;
    if(ch>='a' && ch<='z')
    {
        if(ch+k>'z')
            ch=ch+k-26;
        else
            ch=ch+k;
    }
    if(ch>='A' && ch<='Z')
    {
        if(ch+k>'Z')
            ch=ch+k-26;
        else
            ch=ch+k;
    }
    printf("%c",ch);
    return 0;
}
```

11)

Pooled cab service

Difficulty: easy
Max Points: 10

Description

A company wishes to provide cab service for their N employees. The employees have distance ranging from 0 to N-1. The company has calculated the total distance from an employee's residence to the company, considering the path to be followed by the cab is a straight path. The distance of the company from itself is O. 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 allotted a range of distance. The company wishes to find the distance for the employees who live within the particular distance range.

Write an algorithm to find the distance for the employees who live within the distance range.

Input

The first line of the input consists of three space-separated integers-num, start and end 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 distances of the employees from the company.

Output

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

Constraints

NA

Example

Input:

6 30 50

29 38 12 48 39 55

Output:

38 48 39

Explanation

There are three employees with distances 38, 48 and 39 whose distance from the office lies within the given range.

Public Test Cases

#	Input	Expected Output
1	6 30 50 29 38 12 48 39 55	38 48 39

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int n,r1,r2;
    scanf("%d %d %d",&n,&r1,&r2);
    int a[n],i,t;
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
        if(a[i]<0)
            t=-a[i];
        else
            t=a[i];
        if(t>=r1 && t<=r2)
            printf("%d ",a[i]);
    }
    return 0;
}
```

12)

Kth Shortest Processing Queue

Difficulty: easy
Max Points: 10

Description

A company wishes to modify the technique by which tasks in the processing queue are executed. There are N processes with unique IDs from 0 to N-1. Each of these tasks has its own execution time. The company wishes to implement a new algorithm for processing tasks. For this purpose they have identified a value K By the new algorithm, the processor will first process the task that has the Kth shortest execution time.

Write an algorithm to find the Kth shortest execution time.

Input

The first line of the input consists of two space-separated integers - numTasks and valueK representing the number of tasks (N) and the value K, which is used as reference, respectively.

The second line consists of N space-separated integers - exectuionTimep exectuionTime2, , exectuion7imeN representing the execution times of the tasks.

Output

Print an integer representing the Kth shortest execution time.

Constraints

0 < value K < numTasks < 106

0 < execution time, < 106

0 < l < numTasks

Example

Input

7 5

9 -3 8 -6 -7 18 10

Output

9

Explanation

The sorted list of execution times is [-7, -6, -3, 8, 9,10, 18].

Public Test Cases

#	Input	Expected Output
1	7 5 9 -3 8 -6 -7 18 10	9

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    int n,k;
    scanf("%d %d", &n,&k);
    int a[n],i,j,t;
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-1-i;j++)
        {
            if(a[j]>a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
```

```
                    a[j+1]=t;
            }
        }
    }
    printf("%d",a[k-1]);
}
```

13)

Charlie Magic Mirror

Difficulty: easy
Max Points: 10

Description

Charlie has a magic mirror. The mirror shows right rotated versions of given word.To generate different right-rotations of a word. Write the word in a circle in clockwise order, then start reading from any given character in clockwise order till you have covered all the characters. In the word "sample", if we start with 'P', we get the right rotated word as "plesam". There are six such right rotations of "sample" including itself.

The inputs to the function isSameReflection consists of two string, word1 and word2. The function returns 1 if word1 and word2 are right rotations of the same word and -1 if they are not. Both word1 and word2 will strictly contain characters between 'a'-'z' (lower case letters).

Constraints

NA

Example

INPUT:

plesam

OUTPUT:

1

Explanation

NA

Public Test Cases

#	Input	Expected Output
1	sample plesam	1

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int i,j;
    char s1[100],s2[100],s3[100];
    scanf("%s %s", s1,s2);
    for(i=0,j=0;s2[i]!='\0';i++,j++)
    {
        s3[j]=s2[i];
    }
    for(i=0;s2[i]!='\0';i++,j++)
    {
        s3[j]=s2[i];
    }
    s3[j]='\0';
    j=0;
    for(i=0;s3[i]!='\0';i++)
    {
        if(s3[i]==s1[j])
        {
            j++;
        }
        else
            j=0;
        if(s1[j]=='\0')
        {
            printf("1");
            break;
        }


    }
    if(s1[j]!='\0')
    {
        printf("-1");
    }
    return 0;
}
```

14)

Employee's Rating Point

Difficulty: easy
Max Points: 10

Description

In a company, an employee's rating point (ERP) is calculated as the sum of the rating points given by the employee's manager and HR. The employee rating grade (ERG) is calculated according to the ERP ranges given below.

ERP	ERG
30-50	D
51-60	C
61-80	B
81-100	A

Write an algorithm to find the ERG character for a given employee's ERP.

Input

The input consists of an integer eRP, representing the calculated employee rating point.

Output

Print a character representing the ERG for a given employee's ERP.

Constraints

30 < eRP < 100

Example

Input

64

Output

B

Explanation

64 lies in range 61 to 80, so the grade is B.

Public Test Cases

#	Input	Expected Output
1	64	B

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    int erp;
    scanf("%d", &erp);
    if(erp>=30 && erp<=100)
```

```
    {
        if(erp>=30 && erp<=50)
            printf("D");
        else if(erp<=60)
            printf("C");
        else if(erp<=80)
            printf("B");
        else
            printf("A");
    }
    return 0;
}
```

15)

Special Discount

Difficulty: easy
Max Points: 10

Description

An e –commerce company is planning to give a special discount on all 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 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 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

9–3 8–7–8 18 10

Output

2

Explanation

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

So, the output is 2

Public Test Cases

#	Input	Expected Output
1	7 18 9 -3 8 -7 -8 18 10	2

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int n,i,dis,c=0;
    scanf("%d %d",&n,&dis);
    int order[n];
    for(i=0;i<n;i++)
    {
        scanf("%d",&order[i]);
    if(order[i]>0 && dis%order[i]==0)
            c++;
    }
    printf("%d",c);
}
```

16)

Gift Hampers to Winners

Difficulty: easy
Max Points: 10

Description

The manager of a supermarket wants to organize an event in which he will distribute gift hampers to the winners of the event. The manager has planned in such a way that each customer has to pick the product in a pair and each pair has different types of products. Any two customers can't pick the same type of product pair but the price may be same. There are N types of products and each product has a price. He will offer the gift hampers to those customers for whom the difference of the price of the products pickedby them is equal to the given integer value K.

Write an algorithm to help the Manager find the total numbers of lucky customers who will get the gift hampers.

INPUT:

ProductTypes : an integer representing the types of products(N)

numK, an integer representing the given value K

prices, a list of integers representing the price of the products.

OUTPUT:

return a integer representing the total number of lucky customers who will get the gift hamper.

Constraints

NA

Example

Sample:

INPUT:

6

13

10 15 23 14 2 15

OUTPUT:

3

Explanation

NA

Public Test Cases

#	Input	Expected Output
1	6 13 10 15 23 14 2 15	3

Programming Language: C (GCC 9.2.0)

```c
#include<stdio.h>
int main()
{
    int n, numk;
    int i,j,c=0;
    scanf("%d",&n);//ProductTypes
    scanf("%d",&numk);
    int prices[n];
    for(i=0;i<n;i++)
        scanf("%d",&prices[i]);
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
        {
            if((prices[i]-prices[j]==numk)||(prices[j]-prices[i]==numk))
                c++;
        }
    }
    printf("%d",c);
    return 0;
}
```

17)

Help Jackson

Difficulty: easy
Max Points: 10

Description

Jackson, a math research student, is developing an application on triangles in mensuration. For the two triangles on the application's display, with base and height given, the user must identify the triangle with the largest area. Jackson must now write an algorithm to find the area of the larger triangle.

To find the area of a triangle with base and height given, the following formula is used:

Area of a triangle = (base*height)/2.

Write an algorithm to find the area of the largest triangle.

Input

The first line of the input consists of two space-separated positive integers – base1, height1, representing the base and height of the first triangle.

The second line consists of two space-separated positive integers- base2, height2, representing the base and height of the second triangle.

Output

Print a real number representing the area of the largest triangle rounded up to 2 decimal places.

Constraints

$0 \leq base1, height1, base2, height2 \leq 109$

Example

Input:

5 8

4 11

Output:

22.00

Explanation

Area of the first triangle = 20.000000((5*8)/2).

Area of the second triangle = 22.000000((4*11)/2).

So, the output is 22.00

Public Test Cases

#	Input	Expected Output
1	5 8 4 11	22.00

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    double b1,h1,b2,h2,a1,a2;
    scanf("%lf %lf %lf %lf", &b1,&h1,&b2,&h2);
    a1=(b1*h1)/2;
    a2=(b2*h2)/2;
    if(a1>a2)
```

```
        printf("%0.2lf",a1);
    else
        printf("%0.2lf",a2);
}
```

18)

Christmas Discount

Difficulty: easy
Max Points: 10

Description

An e-commerce company plans to give their customers a discount for the Christmas
holiday. The discount will be calculated on the basis of the bill amount of the order placed. The
discount amount is the product of the sum of all odd digits and the sum of all even digits of the
customer's total bill amount.

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

Input

The input consists of an integer billAmount, representing the total bill amount of a customer.

Output

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

Constraints

O <billAmount≤ 109

Example

Input:

2514795

Output

162

Explanation

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

Even digits in the given number 2514795 are 2, 4. The sum of these even digits is 6.

So, the output is 162.

Public Test Cases

#	Input	Expected Output
1	2514795	162

Programming Language: C (GCC 9.2.0)

```c
#include<stdio.h>
int main()
{
    long int bill;
    int dis,oddsum=0,evensum=0;
    scanf("%ld",&bill);

    while(bill!=0)
    {
        if(bill%10%2!=0)
        oddsum=oddsum+bill%10;
        else
        evensum=evensum+bill%10;
        bill=bill/10;
    }
    if(oddsum==0)
    oddsum=1;
    if(evensum==0)
    evensum=1;

    printf("%d",oddsum*evensum);
    return 0;
}
```

19)

Stock Trader

Difficulty: easy
Max Points: 10

Description

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

$0 < valueK \leq numOfStocks \leq 106$

$0 \leq stocki \leq 106$

$0 \leq i < numOfStocks$

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.

Public Test Cases

#	Input	Expected Output
1	7 5 9 -3 8 -6 -7 18 10	9

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    int n,k;
    scanf("%d %d", &n,&k);
    int a[n],i,j,t;
    for(i=0;i<n;i++)
    {
        scanf("%d",&a[i]);
    }
```

```
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-1-i;j++)
        {
            if(a[j]>a[j+1])
            {
                t=a[j];
                a[j]=a[j+1];
                a[j+1]=t;
            }
        }
    }
    printf("%d",a[k-1]);
}
```

20)

Missing Data

Difficulty: easy
Max Points: 10

Description

A company provides network encryption for secure data transfer. The data string is encrypted prior to transmission and gets decrypted at the receiving end. But due to some technical error, the encrypted data is lost and the received string is different from the original string by 1 character. Arnold, a network administrator, is tasked with finding the character that got lost in the network so that the bug does not harm other data that is being transferred through the network.

Write an algorithm to help Arnold find the character that was missing at the receiving end but present at the sending end.

Input

The input consists of two space-separated strings – stringSentand stringRec, representing the string that was sent through the network, and the string that was received at the receiving end of the network, respectively.

Output:

Print a character representing the character that was lost in the network during transmission and if there is no data loss during transmission then print "NA".

Constraints

NA

Example

Input:

abcdfjgerj abcdfijger

Output:

j

Explanation

The character 'j' at the end of the sent string was lost in the network during transmission.

Public Test Cases

#	Input	Expected Output
1	abcdfjgerj abcdfjger	j

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    char s[1000],r[1000];
    scanf("%s %s", s,r);
    int i;
    for(i=0;s[i]!='\0';i++)
    {
        if(s[i]!=r[i])
        {
            printf("%c",s[i]);
            break;
        }
    }
    if(s[i]=='\0')
        printf("NA");
}
```

21)

Maximum Energy in Science Lab

Difficulty: easy
Max Points: 10

Description

In a science research lab, combining two nuclear chemicals produces a maximum energy that is the product of the energy of the two chemicals. The energy values of the chemicals can be

negative or positive. The scientist wishes to calculate the sum of the maximized energies of the two elements when the reaction happens.

Write an algorithm to find the total energy produced by the chemicals when the reaction happens.

Input The first line of the input consists of an integer numOfChem, representing the number of chemicals (N). The second line consists of N space-separated integers - enerp ener2, , enerN representing the energies of the chemicals.

Output Print an integer representing the total energy produced by the chemicals when the reaction happens.

Constraints

$0 <\_ num0fChem\ 106$

$-106<ener<106$

$0 < i<numOfChem$

Example

Input: 7

9-3 8-6-7 8 10

Output: 19

Explanation

NA

Public Test Cases

#	Input	Expected Output
1	7 9 -33 8 -6 -7 8 10	19

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>
int main()
{
    int n;
    scanf("%d", &n);
    int i,e[n];
    for(i=0;i<n;i++)
        scanf("%d",&e[i]);
    int j,t;
    for(i=0;i<n-1;i++)
    {
        for(j=0;j<n-1-i;j++)
        {
            if(e[j]<e[j+1])
            {
                t=e[j];
                e[j]=e[j+1];
```

```
            e[j+1]=t;
        }
    }
}
printf("%d",e[0]+e[1]);
}
```

22)

Transmit Data

Difficulty: easy
Max Points: 10

Description

A company wishes to transmit data to another server. The data consists of numbers only. To secure the data during transmission, they plan to reverse the data first. Write an algorithm to reverse the data.

Input

The input consists of an integer data, representing the data to be transmitted.

Output

Print an integer representing the given data in reverse form.

Constraints

NA

Example

Input

5783789

Output:

9873875

Explanation

On reversing the given value, the output is 9873875.

Public Test Cases

#	Input	Expected Output

1	5783789	9873875

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    long int data;
    scanf("%ld", &data);
    while(data)
    {
        printf("%d",data%10);
        data=data/10;
    }
}
```

23)

New Year Discount

Difficulty: medium
Max Points: 10

Description

An e-commerce company plans to give their customers a discount for the New Years holiday. 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 on 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 a discount for the given total bill amount.

Input

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

Output

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

Constraints

0 <billAmount ≤ 109

Example

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.

Public Test Cases

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    long int n;
    scanf("%ld", &n);
    int dis=0;
    while(n)
    {
        if((n%10)%2 !=0)
            dis=dis+n%10;
        n=n/10;
    }
    printf("%d",dis);
}
```

24)

Lucky Customer

Difficulty: medium
Max Points: 10

Description

An e-commerce website wishes to find the lucky customer who will be eligible for full value cash back. For this purpose, a number N is fed to the system. It will return another number that is calculated by an algorithm. In the algorithm, a sequence is generated, in which each number is the sum of the two proceeding numbers. Initially the sequence will have two 1's in it. The system will return the Nth number from the generated sequence which is treated as the order ID. The lucky customer will be the one who has placed that order.

Write an algorithm to help the website find the lucky customer.

Input

The input consists of an integer token, representing the number fed to the system (N).

Output

Print an integer representing the order ID of the lucky customer.

Constraints

NA

Example

Input:

8

Output:

21

Explanation

The sequence generated by the system will be 1,1,2,3,5,8,13,21. The 8th number in the sequence is 21. The lucky customer is the one who has placed the order with order ID 21.

Public Test Cases

#	Input	Expected Output
1	8	21

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int n;
    scanf("%d", &n);
    int i,a[n];
    a[0]=a[1]=1;
    for(i=2;i<n;i++)
    {
        a[i]=a[i-1]+a[i-2];
    }
    printf("%d",a[n-1]);
    return 0
}
```

Energy in Science Lab

Difficulty: easy
Max Points: 10

Description

In a science research lab, the combination of two nuclear substances produces an initial energy A. This energy A changes at a consistent rate R every second. The energy gets multiplied by a constant value R every second. The scientist wishes to calculate the energy produced at every second if the reaction is allowed to happen for N seconds.

Write an algorithm to find the energy produced at every second if the reaction is allowed to happen for N seconds.

Input

The input consists of three space-separated integers –intialEnergy, rate and time. representing the initial energy produced on combining the nuclear substances (A), the consistent rate of change (R), and the seconds for which the reaction is allowed to happen (N), respectively.

Output

Print N space-separated integers representing the energy produced at every second if the reaction is allowed to happen for N seconds.

Constraints

$-106 \leq$ intialEnergy, rate ≤ 106

$0 \leq$ time ≤ 100

Example

Input

5 3 3

Output:

5 15 45

Explanation

For N =1, an initial energy of 5 is generated

For N=2, a consistent rate of 3 is multiplied to it, so it becomes 15

For N=3, again 3 is multiplied to the previous energy value, so it becomes 45.

So the out is 5, 15, 45

Public Test Cases

#	Input	Expected Output
1	5 3 3	5 12 45

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int e,r,n,t;
    scanf("%d %d %d", &e,&r,&n);
    int i;
                        // Reading input from STDIN
    for(i=1;i<=n;i++)
    {
        if(i==1)
            t=e;
        else
            t=t*r;
        printf("%d ",t);
    }
}
```

26)

Online Game

Difficulty: easy
Max Points: 10

Description

You are playing an online game. In the game, a numbers is displayed on the screen. In order to win the game, you have to Count the trailing zeros in the factorial value of the given number. Write an algorithm to count the trailing zeros in the factorial value of the given number.

Input

The input consists of an integer num, representing the number displayed on the screen.

Output

Print An integer representing the count of trailing zeros in the factorial of the given numbers.

Note

The factorial of the number is calculated as the product of integer numbers from 1 to num.

Constraints

Example

Explanation

On calculating the factorial of 5, the output is 120 (1 x2x3x4x5). There is only one trailing 0 in 120, So the output is 1.

Public Test Cases

#	Input	Expected Output
1	5	1

Programming Language: C (GCC 9.2.0)

```c
#include <stdio.h>

int main()
{
    int num;
    scanf("%d", &num);                      // Reading input from STDIN
    long int f=1;
    int tzero=0;
    for(int i=1;i<=num;i++)
        f=f*i;
    while(f)
    {
        if(f%10==0)
        {
            tzero++;
        }
        else
            break;
        f=f/10;
    }
    printf("%d", tzero);        // Writing output to STDOUT
}
```

27)

Sales Report

Difficulty: easy
Max Points: 10

Description

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-separatedIntegers-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.

Constraints

NA

Example

Input

3 4

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.

Public Test Cases

#	Input	Expected Output
1	3 4 100 198 333 323 122 232 221 111 223 565 245 764	333 232 764

28)

Count the Misses

Difficulty: easy
Max Points: 10

Description

A company has launched a new text editor that allows users to enter English letters, numbers and whitespaces only. If a user attempts to enter any other type of character, it is counted as a miss.

Given a string of text, write an algorithm to help the developer detect the number of misses by a given user in the given input.

Input

The input consists of a string textInput, representing the text that is entered in the text editor by the user.

Output

Print an integer representing the number of misses by a given user in the given input.

Constraints

NA

Example

Input:

aa a234bc@ sad$ hsagd^

Output:

3

Explanation

The characters that are counted as misses by the editor are ['@','$','^']

Public Test Cases

#	Input	Expected Output
1	aa a234bc@ sad$ hsagd^	3