1. 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. Android, 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 first line of the input consists of a string-string Sent representing the string that was sent through the network, the next line consists of s string –StringRec, representing the string that was received at the receiving end of the network.

   Output

   Print a character representing the character that was lost in the network during transmission.

   Example

   Input:

   abcdfjgerj

   abcdfjger

   Output:

   j

   Note:

   The input strings string Sent and String Rec consists of lowercase and uppercase English alphabets [i.e., a-z, A-Z]

   Explanation:

   The character 'j' at the end of the sent string

   Solutions

```
import java.util.Scanner;
public class Test
{
  static char findLostCharacter(String s1, String s2)
  {
    int StrCodeTotal1 = 0;
    int StrCodeTotal2 = 0;
    int i;
    for(i=0;i < s2.length();i++)
    {
      StrCodeTotal1 += s1.charAt(i);
     StrCodeTotal2 += s2.charAt(i);
    }
```

```
        StrCodeTotal1 += s1.charAt(i);
        int intChar = StrCodeTotal1 - StrCodeTotal2;
        return (char)intChar;
    }

    public static void main(String[] args)
    {
        Scanner ip=new Scanner(System.in);
        String s1 = ip.next();
        String s2 = ip.next();
        char extraChar = findLostCharacter(s1, s2);
        System.out.println(extraChar);

    }
}
```

Sample Input1:

abcdfjgerj

abdfjgerj

Sample output1

c

2. Josh went to the market to buy N apples. He found two shops, shop A and B, where apples were being sold in lots. He can buy any number of the complete lot(s) but not loose apples. He is confused with the price and wants you to figure out the minimum cost to buy exactly N apples. Write an algorithm for Josh to calculate the minimum cost to buy exactly N apples.

Input Format:

The first line of the input consists of an integer – N, representing the total number of apples that Josh wants to buy.

The second line consists of two space-separated positive integers – M1 and P1, representing the number of apples in a lot and the lot's price at shop A, respectively.

The third line consists of two space-separated positive integer

M2 and P2, representing the number of apples in a lot's price at shop B, respectively.

Output Format:

Print a positive integer representing the minimum price at which Josh can buy the apples.

Sample Input:

19

3 10

4 15

Sample Output:

65

Solution:

```
Import java.util.Scanner;
class Main {
public static void main (String [] args) {
        Scanner sc = new Scanner (System.in);
        int n = sc.nextInt();
        int m1 = sc.nextInt();
        int p1 = sc.nextInt();
        int m2 = sc.nextInt();
        int p2 = sc.nextInt();;
int min_cost = -1;
for (int i=0; m1*i <= n; i++) {
int count2 = n - i*m1;
if (count2%m2 == 0) {
int cost = p1 * i + p2 * (count2/m2);
min_cost = (cost < min_cost || min_cost == -1) ? cost : min_cost;
}
}
if (min_cost != -1)
System.out.println(min_cost);
else
System.out.println("Invalid inputs");
}
}
```

3. Euclid, Pythagoras, Pascal and Monte plan to play in a park. It can be considered to be divided into N (rows)* M (Columns) positions. Pascal, Monte and Euclid stand at three difference positions. As Pythagoras is the position of Euclid and Monte from the diagonal of the parallelogram.

   Write an algorithm that will help Pythagoras to decide where he should stand in the park.

   Input

   The first line of the input consists of two space-separated integers-rows and cols representing the number of rows (N) and number of columns (M), respectively.

   The next N lines consists of M symbols representing the park where'+' represents the position where one of the three is standing and '-'represents the position that is empty. There is no space between symbols. The last line consists of four space –separated integers X1,Y1,X2 and Y2 representing the x and y coordinates of Euclid's position and X and Y coordinates of Monte's position, respectively.

   Output

Print two spaces –separated integers representing the X and Y coordinates of the position where Pythagoras should stand to complete the parallelogram.

Note

The symbols of positions In park contain only '+'and '-'; where '+'represents the position where one of the three friends is standing and '-'represents the position that is empty. There is no space between symbols.

Example

Input:

4 8

_ _ _ _ _ _ _ _

_ + _ _ _ _ _ _

_ _ _ _ _ _ _ _

_ + _ _ _ _ + _

2 2 4 7

Output:

2 7

Explanation:

The positions of Euclid and Monte are (2, 2) and (4, 7), respectively. '+' representing the positions where three of them are standing. Pythagoras should stand at position (2, 7) which will form a rectangular shape parallelogram.

4. In a birthday party, the host decided to gift the guest who wins a game in the party. In the game, the host announces an odd number and asks the participants to find all the maximum sized sum free subsets. The participants will then add all the elements of the sum free subsets. A set is a sum free set if no elements of the set is the sum of any two other two elements in the set. An element can also be considered twice to get the sum. The participant is declared a winner if he/she tells the correct total sum of the elements of all the maximal sized sum free sets. He She will get number of gifts equal to the total sum.

Write an algorithm to find the maximum number of gifts a participant can win.

Input

The input consists of an integer N, representing the odd number announced by the host.

Output

Print an integer representing the maximum number of gifts a participant can win.

Constraints

$1 \leq N \leq 10^6$

Note

The output can be large so print the output modulo 1000000007.

Example

Input:

3

Output:

9

Explanation:

Subsets of the given set are:{}, {1},{2},{3},{1,2},{1,3}, {2,3},{1,2,3}

{1, 2} is not sum free because 1 +1 =2

{1, 2, 3} is also not sum free as 1+2 =3

Sum free subsets are: {},{1},{2},{3},{1,3},{2,3}

Maximal sized sum – free sunsets are {1, 3} and {2, 3}

So, the total number of gifts to be awarded by the host = {1+3} + {2+3} =9

5. Each year the city of Two land organizes a Mermaid show. All the coins used in Two land have denominations with squared values (i.e. 1, 4, 9, 16, 25, 36, and so no, with the maximum value coin of 2500). David wishes to see the show this year. He has an infinite number of coins of each denomination with which to pay the entrance fee. But his father will allow him to visit the show only if he can successfully calculate all the various combinations of coins he could use to pay the entry fee in exact change.

Write an algorithm to help David Calculate the number of ways in which he can pay the entry fee.

Input

The input consists of an integer – num, representing the entrance fee that David must pay to see the show (N).

Output

Print an integer representing the number of ways in which David can pay the entrance fee to see the show.

Constraints

$1 \leq num \leq 2500$

Note

The output can be large so print the output modulo 1000000007.

Example

Input:

4

Output:

2

Explanation:

David can pay the entrance fee in the following ways:

1+1+1+ and 4

The output is 2.

Solutions

```
import java.util.Scanner;
public class MyClass
```

```
{
   static int noOfWays(int amount)
   {
   int i,j;
   int n=50;
 int coins[]=new int[n];
 for(i=1;i<=n;i++)
    coins[i-1]=i*i;

   int a[][]=new int[n][amount+1];
 for(j=0;j<n;j++)
 {
  a[j][0]=1;
 }
 for(i=0;i<=amount;i++)
 {
  if(i%coins[0]==0)
    a[0][i]=1;
  else
    a[0][i]=0;
 }
for(i=1;i<n;i++)
 {
  for(j=1;j<=amount;j++)
  {
   if(coins[i]>j)
   {
     a[i][j]=a[i-1][j];
   }
   else
   {

    a[i][j]=a[i-1][j]+a[i][j-coins[i]];
   }
  }
 }
 /*for(i=0;i<n;i++)
 {
```

```
 for(j=0;j<=amount;j++)
 {
  System.out.print(" "+a[i][j]);
 }
 System.out.println();
}*/
return (a[n-1][amount]);
  }
  public static void main(String args[])
  {
     int entryfee;
     Scanner ip=new Scanner(System.in);
     entryfee=ip.nextInt();
     System.out.println(noOfWays(entryfee));
  }
}
```

Sample Input1:

4

Sample Output1:

2

Sample Input2:

10

Sample Output2:

4

Explanation:

sum(1,1,1,1,1,1,1,1,1,1)=10
sum(4,4,1,1) =10
sum(4,1,1,1,1,1) =10
sum(9,1) = 10

So, no of ways to make 10 is 4.

6. Write a program to implement a bubble sort algorithm for sorting the elements of an array.

Input Format:

The first line corresponds to the size of an array.

The second line corresponds to the elements.

Output Format:

Print the element in ascending order.

Sample Input:

6

11 15 26 38 9 10

Sample Output:

9 10 11 15 26 38

Solution:

```
import java.util.*;
 class Main
{
   void bubbleSort(int arr[])
   {
      int n = arr.length;
      for (int i = 0; i < n-1; i++)
         for (int j = 0; j < n-i-1; j++)
            if (arr[j] > arr[j+1])
            {
               int temp = arr[j];
               arr[j] = arr[j+1];
               arr[j+1] = temp;
            }
   }
void printArray(int arr[])
   {
      int n = arr.length;
      for (int i=0; i<n; ++i)
         System.out.print(arr[i] + " ");
      System.out.println();
   }
      public static void main(String args[])
   {
      Scanner sc =new Scanner(System.in);
      int n;
      n=sc.nextInt();
      int arr[] =new int[n];
for(int i=0;i<n;i++){
         arr[i]=sc.nextInt();
      }
      Main ob = new Main();
      ob.bubbleSort(arr);
```

```
        ob.printArray(arr);
    }
}
```

7. The company WebTech is developing a system to analyze user purchasing behavior to provide a better shopping experience. User behavior sequences are encoded to strings of '0' and '1' where '0' represents the product returned. One of the steps is to find the largest substring containing an equal number of 0s and 1s, which would indicate that a product's purchase and return rate are the same. As one of the team's software development engineers, you need to develop a method to process the encoded string and return the largest substring that contains an equal number of 0s and 1s.

Input

The first line of the input consists of a string *input String* ,representing the encoded string depicting the user behavior.

Output

Print a string representing the largest substring that contains an equal number 0s and 1s.

Note

If more than one largest substring possible then output the string that appeared first.

Example

Input:

11101011100

Output:

01011100

Explanation:

The largest substring that contains an equal number of 0s and 1s is '01011100'.

```
import java.util.*;
public class Solution
{
  static void findMaxLength(int nums[])
  {
      HashMap<Integer,Integer> mymap=new HashMap<Integer,Integer>();
      int sum = 0;
      int sday=-1;
      int eday=-1;
      int longest_subarray = 0;
      int temp[]=nums.clone();
      String res="";
      for(int i=0;i<nums.length;i++)
      {
        if(nums[i]==0)
```

```
            nums[i]=-1;
        }
        for(int i=0;i<nums.length;++i)
        {
           sum += nums[i];
           if(sum==0)
           {
              if(longest_subarray < i+1)
                 longest_subarray = i+1;
              sday=0;
              eday=i;
           }
           else if(mymap.containsKey(sum))
           {
              if(longest_subarray < i-mymap.get(sum))
              {
                longest_subarray = i-mymap.get(sum);
                   sday=mymap.get(sum)+1;
                   eday=i;
              }
            }
              else
              {
               mymap.put(sum,i);
              }
              }
         //  System.out.println(sday+" "+eday);
           if(sday!=-1)
           {
           for(int i=sday;i<=eday;i++)
           {
              res+=(char)(temp[i]+48);
           }
           System.out.println(res);
           }
      }
      public static void main(String args[])
      {
```

```
        Scanner sc=new Scanner(System.in);
        String str=sc.next();
        char ch[]=str.toCharArray();
        int N=str.length();
        int nums[]=new int[N];
        for(int i=0;i<N;i++)
          nums[i]=ch[i]-48;

        findMaxLength(nums);
    }
};
```

Sample Input1:

110111000100

Sample output1:

110111000100

Sample Input2:

110110110110

Sample output2:

0110

8. A company Dictionary 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 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 text Input, representing the text that is given as an input to the application by the user.

   Output:

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

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

   textInput is case-sensitive(i.e. cat and CAT are considered as different words not same).It contains lowercase and uppercase English alphabets [ie.a –z, A-Z].

   Example

   Input:

   cat batman latt matter cat matter cat.

Output:

cat matter

Explanation:

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

Solutions

```
import java.util.*;
public class Solution
{
  static void Dictionary(String input)
  {
    HashSet<String> hs=new HashSet<String>();
    HashSet<String> dict=new HashSet<String>();
    /* HashSet does not allow duplicate elements */
    String words[]=input.split(" ");
    /*for(String x:words)
      System.out.println(x);*/
    for(int i=0;i<words.length;i++)
    {
      if(hs.add(words[i])==false)
        dict.add(words[i]);
    }
   /*  System.out.println(hs);
     System.out.println(dict);*/
    for(String x:dict)
      System.out.println(x);
  }
  public static void main(String args[])
  {
    Scanner ip=new Scanner(System.in);
     String str=ip.nextLine();
     Dictionary(str);
  }
};
```

Sample Input1:

cat batman latt matter cat matter cat

Sample Output1:

Cat

matter

9. 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 ifnput 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

Sample Input

8

10 98 3 33 12 22 21 11

Sample Output

10 98 12 22 3 33 21 11

Explanation

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

Solution

```
import java.util.Scanner;
class Main
{
 public static void main(String args[])
{
  Scanner sc = new Scanner(System.in);
 int arr_size = sc.nextInt();
 int []arr = new int[arr_size];
for(int i = 0; i < arr_size; i++)
  {
    arr[i] = sc.nextInt();
  }
  int left = 0;
  int right = arr_size - 1;
  while(left < right)
 {
 while(arr[left]%2 == 0 && left < right)
 {
```

```
    left++;
}
 while(arr[right]%2 == 1 && left < right)
{
  right--;
 }
if(left < right)
{
int temp = arr[left];
arr[left] = arr[right];
          arr[right] = temp;
          left++;
          right--;
       }
   }
for(int i=0;i<arr_size;i++)
  System.out.printf("%d ", arr[i] );
}
}
```

10. A company wishes to find a comprehensive way to arrange their data in the database. They wish to assign an identifier key value to the data. The data consists of a sequence of As, Bs, and Cs. You have been assigned the task of determining the number of possible ways to find the identifier keys. The identifier key is identified as the substrings that have an equal number of As, Bs, and Cs.

    Write an algorithm to count the possible ways to find the identifier key for the given data string.

    Input:

    The first line of input contains a string, representing the data in the form of As, Bs, and Cs.

    Output:

    Print an integer representing the count of maximum possible ways to find the identifier key for the given data string.

    Input:

    The first line of input contains a string representing the data in the form of As, Bs, and Cs.

    Output:

    Print an integer representing the count of maximum possible ways to find the identifier key for the given data string.

    Example:

    Input:

    ABACABAC

    Output:

3

Explanation:

There are three possible ways to find the identifier key, Substrings BAC, CAB, BAC that have an equal number of As, Bs,and Cs.

Solutions

```
import java.util.*;
public class Main {
  private static int getSubstringWithEqualABC(String str)
  {
    HashMap<String, Integer> map = new HashMap<>();
    map.put("A*A", 1);
    int ca = 0, cb = 0, cc = 0;
    int ans = 0;
    for (int i = 0; i < str.length(); i++) {
      if (str.charAt(i) == 'A')
        ca++;
      else if (str.charAt(i) == 'B')
        cb++;
      else
        cc++;
      String key = (ca - cb) + "*" + (ca - cc);
      ans += map.getOrDefault(key, 0);
      map.put(key, map.getOrDefault(key, 0) + 1);
    }
    return ans;
  }
  public static void main(String[] args)
  {
    Scanner ip=new Scanner(System.in);
    String str = ip.next();
    System.out.println(getSubstringWithEqualABC(str));
  }
}
```

Sample Input1:

ABACABAC

Sample Output1:

3

11. A special event is going to be organized on New Year's Eve in an IT park having N adjoining buildings. Being the event organizer, Emma is given the responsibility to display the largest possible rectangular banner of the event on the adjoining walls of the buildings.

Write an algorithm to help Emma find the largest possible rectangular area of the adjoining walls of the buildings where the banner can be displayed.

Input

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

The second line consists of N space-separated integers representing the heights of N buildings.

Output:

Print an integer representing the largest possible rectangular area of the adjoining walls of the buildings where the banner can be displayed.

The width of each building is of unit distance. The thickness of the walls between any two buildings is ignored.

Example:

Input:

5

3 4 7 4 6

Output:

16

Explanation:

The largest possible area the adjoining buildings includes the building at the common height of 4 units.

The width of each building is of unit distance. So, the area is 4 * 4 = 16

Output:

Print an integer representing the largest possible rectangular area of the adjoining walls of the buildings where the banner can be displayed.

Constraints

$1 \leq num\ \mathbf{B} \leq 10^6$

$0 \leq height \leq 10^6$, where height represents the heights of the buildings.

$0 \leq i < numB$

Note

All the building are connected. The width of each building is of unit distance. The thickness of the wall between any two buildings is ignored.

Solutions:

```
import java.util.*;
public class Solution
{
   static int findMin(int arr[],int i,int j)
   {
```

```
        int min=(int)Math.pow(2,32);
        while(i<=j)
        {
           if(min>arr[i])
              min=arr[i];
           i++;
        }
        return min;
    }
    static int largestrectangle(int arr[],int N)
    {
      int max=0,min=0;
      int i,j;
      for(i=0;i<N;i++)
      {
         for(j=i+1;j<N;j++)
         {
            min=findMin(arr,i,j);
            if(max < (min*((j-i)+1)))
            {
               max=min*((j-i)+1);
            }
         }
      }
      return max;
    }
    public static void main(String args[])
    {
      Scanner ip=new Scanner(System.in);
      int N=ip.nextInt();
      int arr[]=new int[N];
      for(int i=0;i<N;i++)
      {
         arr[i]=ip.nextInt();
      }
      System.out.println(largestrectangle(arr,N));
    }
};
```

Sample Input:

5

1 4 7 2 6

Sample Output:

8

12. A digital machine generates binary data which consists of a string of 0s and 1s. A maximum signal M, in the data, consists of the maximum number of either 1s or 0s appearing consecutively in the data but M can't be at the beginning or end of the string. Design a way to find the length of the maximum signal.

Input

The first line of the input consists of an integer N, representing the length of the binary string. The second line consists of a string of length N consisting of 0s and 1s only.

Output

Print an integer representing the length of the maximum signal.

Example

Example 1:

Input

6

101000

Output

1

Explanation

For 101000, M can be 0 at the second index or at the third index so in both cases max length = 1.

Example2:

Input

9

101111110

Output

6

Explanation

For 101111110, M = 111111 so maxlength = 6.

Solution

```
import java.util.*;
 class Main
{
   public static void main(String args[])
   {
      Scanner sc =new Scanner(System.in);
```

```
        String str;
        int size;
    size=sc.nextInt();
        sc.nextLine();
        str=sc.nextLine();
    int m=str.length();
        char[] arr = str.toCharArray();
      int max=0,count=0;
      int flag=0;
     for(int i=0;i<m;i++)
      {
      if(arr[i]=='1')
      {
      count++;
      flag=1;
      }
    else if(arr[i]=='0' && flag==1)
    {

    count=0;
    flag=0;
    }
    if(count>max)
    {
    max=count;
    }
    }
    System.out.println(max);
    }
    }
```

13. A company is transmitting its data to another server. To secure the data against malicious activity, they plan to reverse the data before transmitting. They want to know the number of data character that do not change position even after the data stream is reversed. The network administrator has been tasked with ensuring the smooth transmission of the data.

    Write an algorithm for the network administrator to help him find the number of data characters that do not change position even after the data stream is reversed.

    Input:

The input consists of a string DataStream, representing the data to be transmitted throught the network(N).

Output:

Print an integer representing the number of data character that do not change position even after the data stream is reversed, if no such character is found or the input string is empty the print().

Constraints

$0 \leq$ length of dataStream

Note

The input string dataStream is case sensitive and made up of English letters only. Uppercase character and lowercase character are counted as different.

Example

Input:

alphxxdida

Output:

4

Solutions:

```
import java.util.*;
public class Solution
{
  static int samePosition(String input)
  {
    int count=0;
    int i=0;int j=input.length()-1;
    while(i<j)
    {
      if(input.charAt(i)==input.charAt(j))
        count++;
      i++;
      j--;
    }
    if(input.length()%2==0)
      return count*2;
    else
      return (count*2)-1;
  }
  public static void main(String args[])
  {
    Scanner ip=new Scanner(System.in);
```

```
        String str=ip.nextLine();
        System.out.println(samePosition(str));
    }
};
```

Sample Input1:

alphxxdida

Sample Ouput1:

4

Sample Input2:

abcba

Sample Ouput2:

5

14. Write an algorithm to determine the GCD of N positive integers.

Input

The first line of the input consists of an integer num, representing the number of positive integers (N).

The second line consists of N space-separated integersarro, arr_1, arr.1 representing a list of positive integers.

Output

Print an integer representing the GCD of the given positive integers.

Example

Input:

5

2 4 6 8 10

Output

2

Explanation:

The largest positive integer that divides all the positive integers 2 4 6 8 1 0 without a remainder is 2.

So, the output is 2.

Solution :

```
#include<stdio.h>
intfindGCD(intarr[], int n);
int main()
{
intarr[100], n, i;
scanf("%d", &n);
for(i=0; i<n; i++)
{
```

```
scanf("%d", &arr[i]);
}
printf("%d\n", findGCD(arr, n));
return 0;
}
intfindGCD(intarr[], int n)
{
int result = arr[0];
for (int i=1; i<n; i++)
{
result = gcd(arr[i], result);
}
return result;
}
intgcd(int a, int b)
{
if (a == 0 || b == 0)
{
return 0;
}
if (a == b)
{
return a;
}
if (a > b)
{
returngcd(a-b, b);
}
else
{
returngcd(a, b-a);
}
}
```

15. Dinesh is fond of video games. Due to the pandemic, he designs a video game called Corona world. In this game, the player enters the game with certain energy. The player should defeat all the corona infected zombies to reach the next level. When time increases the zombies will increase double the previous minute. Anyhow the player can manage to fight against all the zombies. In this case, definitely the player cannot achieve the promotion. Hence the player gets a superpower to destroy all

the zombies in the current level when the current game time is a palindrome. Anyhow the player can manage only if he knows the time taken to get the superpower. Help the player by providing the minimum minutes needed to get the superpower by which he can destroy all the zombies. You will be provided with the starting time of the game.

Input Format:

First-line contains a string representing the starting time.

Output:

A string representing the minimum minutes needed to get the superpower.

Constraints:

Input time will be in 24 hours format

Sample Input:

05:39

Sample Output:

11

Explanation:

It takes 11 minutes for minute value to become 50, 05:50 is a palindromic time.

Solution

```
import java.util.*;
public class Main
{
  public static void main(String[] a)
  {
Scanner sc = new Scanner(System.in);
String str = sc.nextLine();
int hh,mm;
hh=(str.charAt(0)-48)*10 +str.charAt(1)-48;
mm=(str.charAt(3)-48)*10 +str.charAt(4)-48;
int requiredTime = 0;
while (hh % 10 != mm / 10 || hh / 10 != mm % 10)
    {
     ++mm;
     if (mm == 60)
     {
       mm = 0;
       ++hh;
     }
      if (hh == 24)
         hh = 0;
```

```
        ++requiredTime;
      }
      System.out.println(requiredTime);
    }
}
```

16. Study the graphs carefully and answer the questions that follow.

Monthly expenditure (in dollars) of four persons

Mark, john, james and scott……

Problem statement

Excel columns are labeled in alphabetical order, i.e., A, B, C….Z, AA, AB….AZ, BA, BB….BZ, CA….ZA…… ZZ, AAA, AAB…AAZ and so on. The column index is 1- based i.e. A is represented by 1, 8 is represented by 2, AA is represented by 27, AB by 28, and so on.

Note:

The indexing starts at 1.

The corresponding column 'label' of **n** should be in the upper case.

Input format:

The input consists of a single line: ate.

The line contains a single integer. I.e. **n**.

Input will be read from the STDIN by the candidate.

Output Format

Print a string consisting of the corresponding column 'label' of the number **n.**

The output will be matched to the candidate's output printed on the STDOUT

Constraints:

$0< N < 10^9$

Example:

Input:

956

Output:

AJT

Solution :

```
#include <stdio.h>
voidprintString(int n)
{
intarr[10000];
int i = 0;
while (n) {
arr[i] = n % 26;
n = n / 26;
```

```
i++;
}
for (int j = 0; j < i - 1; j++) {
if (arr[j] <= 0) {
arr[j] += 26;
arr[j + 1] = arr[j + 1] - 1;
}
}
for (int j = i; j >= 0; j--) {
if (arr[j] > 0)
printf("%c", (char)('A' + arr[j] - 1));
}
printf("\n");
}
int main()
{
int n;
scanf("%d",&n);
printString(n);
return 0;
}
```

17. **Problem statement**

You are given string str of length **N.** you are required to find the number of unique characters in string str and print the same.

Note:

If str = NULL, print -1.

Input format:

The input consists of two lines:

The first line contains an integer denoting **N.**

The second line contains a string denoting str.

Input will be read the STDIN by the candidate

Output format:

The output consists of a single integer denoting the number of unique character is str

The output will be matched to the candidates output printed on the stdout

Constraints:

$0 \leq N \leq 10^5$

Str contains only lower case alphabets.

Example:

Input:

7

Abcdabc

Output:

4

Explanation:

The string str is "abcdabc", and unique characters in string str are {'a', 'b', 'c', 'd' }, count of them is4,hence 4 is returned.

Sample input

8

Efeefhjg

Sample output

5

Instructions:

Program should take input from standard input and print output to standard output.

Your code is judged b y an automated system, do not write any additional welcome/greeting messages.

"Save and test" only checks for basic test cases, more rigorous cases will be used to judge your code while scoring.

Additional score will be given for writing optimized code both in terms of memory and execution time.

Solution :

```
#include<stdio.h>
#include<string.h>
intcount_unique_char(char* str) {
int hash[128] = { 0 };
int i, c = 0;
for (i = 0; i <strlen(str); ++i) {
hash[str[i]] = 1;
}
for (i = 0; i < 128; ++i) {
c += hash[i];
}
return c;
}
int main() {
charstr[300];
printf("Enter String: ");
```

```
gets(str);
printf("Number of Unique Characters in String: %d", count_unique_char(str));
}
```

18. Given an integer matrix of size N x N. Traverse it in a spiral form.

Format:

Input:

The first line contains N, which represents the number of rows and columns of a matrix. The next N lines contain N values, each representing the values of the matrix.

Output:

A single line containing integers with space, representing the desired traversal.

Constraints:

0 < N < 500

Example:

Input:

3
1 2 3
4 5 6
7 8 9

Output:

1 2 3 6 9 8 7 4 5

Solution:

```
import java.util.ArrayList;
import java.util.List;
import java.util.Scanner;
public class Main {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
int matrix[][] = new int[n][n];
for (int i = 0; i < n; i++) {
for (int j = 0; j < n; j++)
matrix[i][j] = sc.nextInt();
}
List<Integer> ans = new ArrayList<>();
int totalSize = n * n;
int top = 0, bottom = n - 1, left = 0, right = n - 1;
while (ans.size() < totalSize) {
for (int i = left; i <= right && ans.size() < totalSize; i++)
```

```
ans.add(matrix[top][i]);
top++;
for (int i = top; i <= bottom && ans.size() < totalSize; i++)
ans.add(matrix[i][right]);
right--;
for (int i = right; i >= left && ans.size() < totalSize; i--)
ans.add(matrix[bottom][i]);
bottom--;
for (int i = bottom; i >= top && ans.size() < totalSize; i--)
ans.add(matrix[i][left]);
left++;
}
for (int element : ans)
System.out.print(element + " ");
}
}
```

19. You are given an integer array arr of stre N. Your task is to find and print the minimum element of the array along with the index.

Note:

Array index starts with 0

• The minimum element and its index should be separated by a line in the output . Assume there is only I minimum element in the array

• Print exactly what is asked, do not print any additional greeting messages

Input Format:

The input consists of two lines:

The first line contains N.

The second line contains arr.

Input will be read from the STDIN by the candidate

Output Format:

The output consists of two lines:

Print the minimum element in the first line.

Print the index of the minimum element in the second line. The output will be matched to the candidate's output printed on the STDOUT

Constraints:

ISN≤100

-105 Sarr[0] ≤ 105

Example:

Input: 10

23 45 82 27 66 12 78 13 71 86

Output:

12

5

Explanation:

12 is the minimum element of the array at index 5.

Solution :

```
#include<stdio.h>
int main()
{
int n;
scanf("%d",&n);
intarr[n];
for(int i=0;i<n;i++)
scanf("%d",&arr[i]);
int min=arr[0],idx=0;
for(int i=1;i<n;i++)
if(arr[i]<min)
  {
min=arr[i];
idx=i;
  }
printf("%d is the minimum element of the array at index %d",min,idx);
}
```

20. **Sum of divisors**

Problem statement

You are given an integer n find and print the sum of all of its divisors starting from 1.

Note:

Sum lies within the integer range.

Input Format:

The input consists of a single line of input: . The line contains a single integer, i.e. n.

Input will be read from the STDIN by the candidate

Output Format:

The output will be a single integer, i.e. the sum of all its divisors starting from 1. The output will be matched to the candidate's output printed on the STDOUT

Constraints:

0 <n<10

Example:

Input:

6

Output:

12

Explanation:

Divisors of 6 are (1, 2, 3, and 6)

Sum = 1 + 2 + 3 + 6 = 12, hence the output is 12

Solution :

```
#include<stdio.h>
int main()
{
intn,d=0;
scanf("%d",&n);
for(int i=1;i<=n;i++)
if(n%i==0)
    d=d+i;
printf("%d",d);
}
```