1)You will be given an array of integers. All of the integers except one occur twice. That one is unique in the array.

Given an array of integers, find and print the unique element.

For example **a=[1,2,3,4,3,2,1]** the unique element is **4**.

Function Description

Complete the *lonelyinteger* function in the editor below. It should return the integer which occurs only once in the input array.

lonelyinteger has the following parameter(s):

- *a*: an array of integers

Input Format

The first line contains a single integer **n** denoting the number of integers in the array.
The second line contains **n** space-separated integers describing the values in **a**.

Output Format

Print the unique integer in the array.

Sample Input 0

1

1

Sample Output 0

1

Java

```
import java.util.*;

class Lonely{

            public static void main(String[] args) {

                        Scanner sc=new Scanner(System.in);
```

```
                int a[]=new int[100];
                System.out.println("Enter range");
                int n=sc.nextInt();


                for(int i=0;i<n;i++)
                        a[i]=sc.nextInt();
                System.out.println("Lonely elements are :");
                Lonely.getLonely(a,n);


        }
        public static void getLonely(int a[],int n)
        {
                int pair=0,count=0;


                for(int i=0;i<n;i++)
                        {
                         for (int j=0;j<n ;j++ )
                          {
                                  if(a[i]==a[j])
                                          count++;
                          }
                          if(count == 1)
                          System.out.println(a[i]);
                        count=0;
                }
         }
}

         }
}
```

2)Two cats and a mouse are at various positions on a line. You will be given their starting positions. Your task is to determine which cat will reach the mouse first, assuming the mouse doesn't move and the cats travel at equal speed. If the cats arrive at the same time, the mouse will be allowed to move and it will escape while they fight.

You are given **q** queries in the form of **x ,y** and ,**z** representing the respective positions for cats **A** and **B** and for mouse **C** .

Complete the function **catAndMouse** to return the appropriate answer to each query, which will be printed on a new line.

- If cat **A**catches the mouse first, print Cat A.
- If cat **B**catches the mouse first, print Cat B.
- If both cats reach the mouse at the same time, print Mouse Cas the two cats fight and mouse escapes.

For example, cat **A** is at position **x=2** and cat **B** is at **y=5** . If mouse **C** is at position **z=4**,It is **2** units from cat **A** and **1** unit from cat **B**. Cat **B** will catch the mouse.

Function Description

Complete the *catAndMouse* function in the editor below. It should return one of the three strings as described.

catAndMouse has the following parameter(s):

- *x*: an integer, Cat **A**'s position
- *y*: an integer, Cat **B'**s position
- *z*: an integer, Mouse **C**'s position

Input Format

The first line contains a single integer **q**, denoting the number of queries.
Each of the **q** subsequent lines contains three space-separated integers describing the respective values of **x** (catA 's location) ,**y** (catB 's location), and **z** (mouse C's location).

Output Format

For each query, return Cat A if cat A catches the mouse first, Cat B if cat B catches the mouse first, or Mouse C if the mouse escapes.

Sample Input 0

2

1 2 3

1 3 2

Sample Output 0

Cat B

Mouse C

Java

```
import java.util.*;
class CatandMouse{


public static void main(String[] args) {
            Scanner sc=new Scanner(System.in);
            System.out.println("Enter no of queries");
            int q=sc.nextInt();
            for (int i=0; i<q;i++ ) {
                        CatandMouse.createAarry();
            }
 }
            public static void createAarry()
            {
                        Scanner sc=new Scanner(System.in);
                        int a[]=new int[3];
                        System.out.println("Enter queries");
                        for (int i=0;i<a.length;i++) {
                                    a[i]=sc.nextInt();
```

```
                    }
                    if( Math.abs(a[2]-a[1]) < Math.abs(a[2]-a[0]) )
                                System.out.println("Cat B catches first");
                    if( Math.abs(a[2]-a[1]) > Math.abs(a[2]-a[0]) )
                                System.out.println("Cat A catches first");
                    if( Math.abs(a[2]-a[1]) == Math.abs(a[2]-a[0]) )
                                System.out.println("Mouse C can escape");
          }
}
```

3)Monica wants to buy a keyboard and a USB drive from her favorite electronics store. The store has several models of each. Monica wants to spend as much as possible for the **2** items, given her budget.

Given the price lists for the store's keyboards and USB drives, and Monica's budget, find and print the amount of money Monica will spend. If she doesn't have enough money to both a keyboard *and* a USB drive, print -1 instead. She will buy only the two required items.

For example, suppose she has **b=60** to spend. Three types of keyboards cost **keyboards=[4,50,60]**. Two USB drives cost **drives=[5,8,12]**. She could purchase a **40 keyboards + 12 USB drives =52** or a **50 keyboards + 8 USB drives =58**. She chooses the latter. She can't buy more than **2** items so she can't spend exactly **60**.

Function Description

Complete the *getMoneySpent* function in the editor below. It should return the maximum total price for the two items within Monica's budget, or **-1** if she cannot afford both items.

getMoneySpent has the following parameter(s):

- *keyboards*: an array of integers representing keyboard prices
- *drives*: an array of integers representing drive prices
- *b*: the units of currency in Monica's budget

Input Format

The first line contains three space-separated integers **b ,n**, and **m** , her budget, the number of keyboard models and the number of USB drive models.
The second line contains **n** space-separated integers **keyboard[i]**, the prices of each keyboard model.
The third line contains **m** space-separated integers **drives**, the prices of the USB drives.

Output Format

Print a single integer denoting the amount of money Monica will spend. If she doesn't have enough money to buy one keyboard *and* one USB drive, print -1 instead.

Sample Input 0

10 2 3

3 1

5 2 8

Sample Output 0

9

Java

```
import java.util.*;
class sahil{
            public static void main(String[] args) {
                        Scanner sc=new Scanner(System.in);
                        int keboard[]=new int[100];
                        int usb[]=new int[100];
                        int a[]=new int [100];
                        int n=keboard.length;
                        int m=usb.length;
                        System.out.println("Enter rupees");
                        int s=sc.nextInt();
                        System.out.println("Enter no.of keyboards");
                        n=sc.nextInt();
                        System.out.println("enter no of usbs");
                        m=sc.nextInt();
                        System.out.println("Enter brand of keboards");
                        for (int i=0; i<n;i++ ) {
                                    keboard[i]=sc.nextInt();
                        }
                        System.out.println("Enter usb brands");
                        for (int i=0;i
                                    usb[i]=sc.nextInt();
                        }
                        int max=-1;
        for(int i=0;i<n;i++)
        {
            for(int j=0;j<m;j++)
```

```
            {
                        if(keboard[i]+usb[j]<=s && keboard[i]+usb[j]>max)
                                    max=keboard[i]+usb[j];
            }
        }
        System.out.println(max);
            }
}
```

4)You are in charge of the cake for your niece's birthday and have decided the cake will have one candle for each year of her total age. When she blows out the candles, she'll only be able to blow out the tallest ones. Your task is to find out how many candles she can successfully blow out.

For example, if your niece is turning **4** years old, and the cake will have **4** candles of height **4 ,1 ,1 ,3** she will be able to blow out **2** candles successfully, since the tallest candles are of height **4** and there are **2** such candles.

Function Description

Complete the function birthdayCakeCandles in the editor below. It must return an integer representing the number of candles she can blow out.

birthdayCakeCandles has the following parameter(s):

- *ar*: an array of integers representing candle heights

Input Format

The first line contains a single integer **n** denoting the number of candles on the cake.
The second line contains **n** space-separated integers, where each integer **i** describes the height of candle **i** .

Output Format

Return the number of candles that can be blown out on a new line.

Sample Input 0

4

3 2 1 3

Sample Output 0

2

Java

```
import java.util.*;

class BirthdayCandels{
```

```
	public static void main(String[] args) {
		Scanner sc=new Scanner(System.in);
		int candels[]=new int[100];
		System.out.println("Enter range");
		int n=sc.nextInt();
		System.out.println("Enter candels");
		for (int i=0; i<n;i++ ) {
		   candels[i]=sc.nextInt();
		}
		int max=0,count=0;
		max=candels[0];
		for (int i=1;i<n;i++ ) {
			if(candels[i]>max )
				max=candels[i];
		}
			for (int j=0;j<n ; j++) {
				if(max==candels[j])
					count++;
			}
	System.out.println(count);
	}
}
```

5)Gary is an avid hiker. He tracks his hikes meticulously, paying close attention to small details like topography. During his last hike he took exactly **n** steps. For every step he took, he noted if it was an *uphill* **U**, or a *downhill*, **D** step. Gary's hikes start and end at sea level and each step up or down represents a 1 unit change in altitude. We define the following terms:

- A *mountain*is a sequence of consecutive steps *above* sea level, starting with a step *up* from sea level and ending with a step *down* to sea level.
- A *valley*is a sequence of consecutive steps *below* sea level, starting with a step *down* from sea level and ending with a step *up* to sea level.

Given Gary's sequence of *up* and *down* steps during his last hike, find and print the number of *valleys* he walked through.

For example, if Gary's path is **s=[DDUUUUDD]** he first enters a valley 2 units deep. Then he climbs out an up onto a mountain 2 units high. Finally, he returns to sea level and ends his hike.

Function Description

Complete the *countingValleys* function in the editor below. It must return an integer that denotes the number of valleys Gary traversed.

countingValleys has the following parameter(s):

- *n*: the number of steps Gary takes
- *s*: a string describing his path

Input Format

The first line contains an integer n , the number of steps in Gary's hike.
The second line contains a single string **s** of **n** characters that describe his path.

Print a single integer that denotes the number of valleys Gary walked through during his hike.

Sample Input

8

UDDDUDUU

Sample Output

1

Java

```
class Valley{

            public static void main(String[] args) {

                        String s="DUDUDDDUU";

                        int sealevel=0,valley=0;

                        for(int i=0;i<s.length();i++){

                                    char ch=s.charAt(i);

                                    if(ch=='U')

                                                sealevel++;

                                    else{



                                                if(sealevel==0)

                                                            valley++;

                                                sealevel--;

                                    }

                        }

System.out.println(valley);

            } }
```

6)John works at a clothing store. He has a large pile of socks that he must pair by color for sale.

Given an array of integers representing the color of each sock, determine how many pairs of

socks with matching colors there are.

Function Description:

Complete the sockMerchant() function in the editor below. It must return an integer representing the number of matching pairs of socks that are available.

sockMerchant has the following parameter(s):

n

: the number of socks in the pile

-

ar

: the colors of each sock.

Input Format

The first line contains an integer

, the number of socks represented in

.

The second line contains

space-separated integers describing the colors

of the socks in the pile.

Output Format

Return the total number of

matching pairs

of socks that John can sell.

Sample Input

9

10 20 20 10 10 30 50 10 20

Sample Output

3

Java

```
class Socs{

            public static void main(String[] args) {

             int ar[]={1,2,1,2,1,3,2};

             int pair=0,count=0;

             for(int i=0;i<ar.length;i++)

              {

                        for(int j=i;j<ar.length;j++)

                                    {

                                                if(ar[i]==ar[j])

                                                            count++;

                                    }

                                    if(count%2==0)

                                                pair++;
```

```
                        count=0;

                }

                System.out.println(pair);

            }

}
```

7)Write a program to check whether a given number is a happy number or unhappy number.

Happy number: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1, or it loops endlessly in a cycle which does not include 1.

Java

```
import java.util.*;

public class Main

{

                public static void main(String[] args)

                {

                                Scanner sc=new Scanner(System.in);

                    int n,s=0,a=0;

        n=sc.nextInt();

                    while(n>9)

                  {

                                while(n!=0)

         {

                                                a=(int)Math.pow(n%10,2);

                                                s=s+a;

                                                    n=n/10;

         }

         n=s; s=0;

     }

     if(n==1)

System.out.println("Happy no.");

    else

System.out.println("Not Happy no.");

                }

}
```

We define the distance between two array values as the number of indices between the two values.

Given a, find the minimum distance between any pair of equal elements in the array. If no such value exists, print-1.

Python

```
import sys


n = int(raw_input().strip())
A = map(int,raw_input().strip().split(' '))
r=n+1
for i in range(n):
for j in range(i+1,n):
if A[j]==A[i]:
r=min(j-i,r)
if r<=n:
print r
else:
print -1
```

8)Monica wants to buy a keyboard and a USB drive from her favorite electronics store. The store has several models of each. Monica wants to spend as much as possible for 2 the items, given her budget.

Given the price lists for the store's keyboards and USB drives, and Monica's budget, find and print the amount of money Monica will spend. If she doesn't have enough money to both a keyboard *and* a USB drive, print -1 instead. She will buy only the two required items.

Python

```
import sys


s,n,m = input().strip().split(' ')
s,n,m = [int(s),int(n),int(m)]
k = [int(keyboards_temp) for keyboards_temp in input().strip().split(' ')]
p = [int(pendrives_temp) for pendrives_temp in input().strip().split(' ')]
ans = -1
for i in k:
for j in p:
if i + j <= s and i + j >= ans:
ans = i + j
print(ans)
```

9)Larry has been given a permutation of a sequence of natural numbers incrementing 1 from as an array. He must determine whether the array can be sorted using the following operation any number of times:

- **Choose any 3 consecutive indices and rotate their elements in such a way thatABC-> BCA->CAB->ABC.**

C / C++

```
#include "bits/stdc++.h"
using namespace std;
const int N = 1e3 + 3;
int t;
int n;
int arr[N];
int inv;
int main(){
cin>> t;
while(t--){
cin>> n;
for(int i = 1 ; i<= n ; ++i){
cin>>arr[i];
}
inv = 0;
for(int i = 1 ; i<= n ; ++i){
for(int j = i + 1 ; j <= n ; ++j){
inv += (arr[j] <arr[i]);
}
}
if(inv & 1){
cout<< "NO\n";
}
else{
cout<< "YES\n";
}
}
}
```

10)An integer d is a *divisor* of an integer n if the remainder of n/d = 0.

Given an integer, for each digit that makes up the integer determine whether it is a divisor. Count the number of divisors occurring within the integer.

Note: Each digit is considered to be unique, so each occurrence of the same digit should be counted (e.g. for n=111,1is a divisor of 111 each time it occurs so the answer is3).

C / C++

```
#include <stdio.h>

int main()
{
int T;
scanf("%d",&T);
while(T--)
{
int N,M,d,c=0;
scanf("%d",&N); M=N;
while(N)
{
d=N%10;
N=N/10;
if(d &&M%d==0) c++;
}
printf("%d\n",c);
}
return 0;
}
```

11)John Watson knows of an operation called a *right circular rotation* on an array of integers. One rotation operation moves the last array element to the first position and shifts all remaining elements right one. To test Sherlock's abilities, Watson provides Sherlock with an array of integers. Sherlock is to perform the rotation operation a number of times then determine the value of the element at a given position.

For each array, perform a number of right circular rotations and return the value of the element at a given index.

For example, array a=[3,4,5], number of rotation, k=2 and indices to check, .
m=[1,2] First we perform the two rotations:

[3,4,5]à [5,3,4]à [4,5,3]

Now return the values from the zero-based indices 1 and 2 as indicated in the m array.

a[1]=5

a[2]=3

C / C++

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;


int main() {
int n, k, q;
scanf("%d", &n);
scanf("%d", &k);
scanf("%d", &q);
int data[n];
for(int i=0; i<n; i++) scanf("%d", &data[i]);
k = k % n;
while(q--) {
int x;
scanf("%d", &x);
x = x - k;
if(x < 0) x = x + n;
printf("%d\n", data[x]);
}
return 0;
}
```

12)The Utopian Tree goes through *2* cycles of growth every year. Each spring, it *doubles* in height. Each summer, its height increases by *1* meter.

Laura plants a Utopian Tree sapling with a height of *1* meter at the onset of spring. How tall will her tree be after

growth cycles?

For example, if the number of growth cycles is n=5, the calculations are as follows:

Periods Height

- **1**
- **2**
- **3**
- **6**
- **7**
- **14**

Python

```
for i in range(input()):

n=input()

h=1

for i in range(1,n+1):

        if i&1:

              h*=2

       else:

             h+=1

print h
```

13)Anna and Brian are sharing a meal at a restaurant and they agree to split the bill equally. Brian wants to order something that Anna is allergic to though, and they agree that Anna won't pay for that item. Brian gets the check and calculates Anna's portion. You must determine if his calculation is correct.

For example, assume the bill has the following prices:bill=[2,4,6]. Anna declines to eat item k=bill[2] which costs 6. If Brian calculates the bill correctly, Anna will pay (2+4)/2=3. If he includes the cost of bill[2],he will calculate (2+4+6)/2 = 6. In the second case, he should refund 3 to Anna.

C / C++

```
#include <cmath>
#include <cstdio>
#include <vector>
#include <iostream>
#include <algorithm>
using namespace std;




int main() {
    int n, k, sum=0;
cin>> n >> k;
    for (int i=0;i<n;i++) {
        int a;
cin>> a;
        if (i!=k) sum+=a;
    }
    int l;
cin>> l;
    if (sum/2==l) cout<< "Bon Appetit" <<endl;
    else cout<< l-sum/2 <<endl;
}
```

14)A discrete Mathematics professor has a class of students. Frustrated with their lack of discipline, decides to cancel class if fewer than some number of students are present when class starts. Arrivel times go from on time (arrival time<=0) to arrived late (arrivalTime> 0).

Given the arrival time of each student and a threshold number of attendees, determine if the class is canceled.

C / C++

```
#include<iostream>
                #include<stdio.h>
                #include<algorithm>
#include<string>
#include<vector>
                #include<math.h>
                Using namespce std;
                int main() {
                                ios : : sync_with_stdio(0);
                                int t,n,k,x;
                                cin>> t;
                                while( tyà 0) {
                                                cin>> n >> k;
                                while(n à 0) {
                                                cin>> x;
                                                k -=( x<=0);
                                }
                                If ( k> 0 )
                                                Cout<, “YES\n”;
                                else
                                                Cout<< “NO\n”;
                                }
                                                return 0;
```

15)You have been asked to help study the population of birds migrating across the continent. Each type of bird you are interested in will be identified by an integer value. Each time a particular kind of bird is spotted, its id number will be added to your array of sightings.

You would like to be able to find out which type of bird is most common given a list of sightings. Your task is to print the type number of that bird and if two or more types of birds are equally common, choose the type with the smallest ID number.

For example, assume your bird sightings are of types arr = [1,1,2,2,3]. There are two each of types 1 and 2, and one sighting of type 3. Pick the lower of the two types seen twice: type 1.

C / C++

```
#include<iostream>
                Using namespace std;
                Int cnt[6];
                Int main() {
                Int n;
                Cin>> n;
                For ( inti = 0; I < n;  i++) {
                Int x;
                Cin>> x;
                Cnt[x]++;
                }
                int mx =1;
                for( inti =2; i< 6; i++) {
                                if( cnt[i] >cnt[mx]) {
                                mx = i;
                                }
                }
                Cout<< mx << end1;
                return  0;
}
```

16)You will be given two array of integers and asked to determine all integer that satisfy the following two conditions:

1: The elements of the first array are all factors of the integer being considered

2: The integer being considered is a factor of all elements of the second array.

These number are referred to as being between the two arrays. You must determine how much such number exists.

Python

```
Import sys
                n,m=raw_input().strip().split(‘ ‘)
                n,m=[ int(n),int(m)]
                a= map(int,raw_input().strip().split(‘ ‘))
                b= map(int,raw_input().strip().split(‘ ‘))
                cnt  = 0
                for I n range(1, 101 ):
                                works = True
                                for j in a:
                                      if i % j ! = 0:
                                                works = false
                                for j in b:
                                      if j % I ! = 0:
                                                works  = false
                                if works :
                                                cnt += 1
                                print cnt
```

17)Given a square matrix. Calculate the absolute difference between the sums of its diagonals.

C / C++

```
#include<iostream>

Using namespace std;

Int main()

Int n;

Cin>> n;

Int arr[n][n];

Long long int d1=0; // First diagonal

Long long int d2=0; // Second diagonal

For(int i=0; i<n; i++) {

For(int j=0;j<n;j++) {

Cin>>arr[i][j];

If(i==j) d1 += arr[i][j];

If(i==n-j-1) d2 += arr[i][j];

}

}

Cout<< abs(d1-d2) << end; // absolute difference of the sum across the diagonals

return 0;

}
```

18)You are choreographic a circus shows with various animals. For one act, you are given two kangaroos on a number line ready to jump in the positive (i.e towards positive infinity).

- **The first kangaroos starts at location x1 and moves at a rate of v1 meter per jump.**
- **The second kangaroos starts at location x2 and moves at a rate of v2 meter per jump.**

You have to figure out a way to get both kangaroos at the same location at the same time as part of the shows .If it is possible , return YES, otherwise return NO.

For example: – kangaroos 1 starts at x1=2 with a jump distance v1=1 and kangaroos 2 starts at x2=1 with a jump distance of v2. After on jump, they are both at x=3. (x1+v1=2 x2+v2=1+2), So our answer is YES.

Python

```
X1,v1,x2,v2 =map(int,raw_input(),split())
    X= [x1,v1]
    Y= [x2,v2]
    Back= min(x,y)
    Fwd= max(x,y)
    Dist= fwd[0]-back[0]

    while back[0] <fwd[0];
        back[0] += back[i]
        fwd[0] += fwd[1]
    if fwd[0] – back[0]>= dist :
        break
    print [“ NO” , “YES”] [back[0] == fwd[0]
```

19)Given five positives, Find the minimum and maximum values that can be calculated by summing
Exactly four the five integers. Then print the respective minimum and maximum values as a single
Line of two space- separated long integer

Java

```
Import java.util.*;

                Public class Solution {

                                Public static void main(String[] args) {

                                                Scanner scan = new Scanner (System.in);

                                                Long sum=0;

                                                Long min=1000000000;

                                                Long max=0;

                                While( Scan.hasNext()) {

                                                Long n =scan.nextLong();

                                                Sum = sum+ n;

                                                If (min > n) {

                                                    Min =n;

                                                }

                                                If (max < n) {

                                                max = n;

                                }

                }

                Scan.close();

                System.out.println(sum-max) + " " +(sum- min));

                }

}
```

20)Given an array of integer, Find the sum of its elements.
For example, If the array ar = [1,2,3], 1+2+3=6, so return 6.

C / C++

```
#include

int main()

{

int n;

int i;

scanf(“%d”, &n);

int array[n];

int sum_of_array = 0;

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

scanf(“%d”, &array[i]);

sum_of_array += array[i];

}

printf(“%d\n”,sum_of_array);

return 0;

}
```

21)Write a Program for Matrix Multiplication(Universal Program)

```
#include <stdio.h>
void main()
{
int arr[2][2],arr2[2][2];
int temp,i,j;
printf(“Enter elements for 1st array\n”);
for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
scanf("%d",&arr[i][j]);
}
}
printf("Enter elements for 2nd array\n");
for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
scanf("%d",&arr2[i][j]);
}
}
printf("elements for 1st and 2nd array\n");
for(i=0;i<2;i++)
{
for(j=0;j<2;j++)
{
printf("%d \t",arr[i][j]);
}
printf("\t");
for(j=0;j<2;j++)
{
printf("%d \t",arr2[i][j]);
}
printf("\n");
}
printf("\n");
for(i=0;i<2;i++)
{

for(j=0;j<2;j++)
{
for(int k=0;k<2;k++)
{
temp=temp+((arr[i][k])*(arr2[k][j]));

}
printf("%d\t",temp);
temp=0;

}

printf("\n");
}

}
```

22)Write a Program to remove vowels from a string?

```
#include <stdio.h>
#include <string.h>

int check_string(char);

int main()
{
char string[100], tp[100];
int c, d = 0;

printf("Enter a string you want to manipulate: ");
gets(string);

for(c = 0; string[c] != '\0'; c++) {
if(check_string(string[c]) == 0) {
tp[d] = string[c];
d++;
}
}

tp[d] = '\0';

strcpy(string, tp);

printf("The string after deletion of vowels: %s\n", string);

return 0;
}

int check_string(char ch)
{
if (ch == 'a' || ch == 'A' || ch == 'e' || ch == 'E' || ch == 'i' || ch == 'I' || ch =='o' ||
ch=='O' || ch == 'u' || ch == 'U')
return 1;
else
return 0;
}
```

23)Write a function that takes an array called scores and 2 pointer parameters min and max and determines the minimum and maximum values and returns them to the calling function

```
#include <stdio.h>
void minmax(int score[],int len,int *min,int *max)
{
int i;
*max = score[0];
*min = score[0];
for(i = 0 ; i < len ; i++)
{
if(*max < score[i])
*max = score[i];
if(*min > score[i])
*min = score[i];
}
}

int main()
{
int score[100],i,len,min,max;
printf("\nPlease enter the length of array:");
scanf("%d",&len);
for(i = 0 ; i < len ; i++)


printf("Enter the a[%d] values: ",i);
scanf("%d",&score[i]);
}
minmax(score,len,&min,&max);
printf("\nMaximum values is: %d \nMinimum value is: %d",max,
return 0;
}
```

24)Write a function that uses character handling library functions to determine the number of upper case, lower case, digits, spaces and punctuation characters in the specified text

```
#include <stdio.h>
#include <string.h>
#include <math.h>

int main()
{
int length, i;
int alpha = 0, punct = 0, digit = 0, lower = 0, upper = 0,
spaces = 0, others = 0;
char buffer[150] = "English Language consists of letters
from A to Z and digits 1 to 9 which are
used to form : words, sentences and
paragraphs!!!";
length = strlen(buffer);
for ( i=0; i<length; i++ )
{
if( isalpha( buffer[i] ))
{
alpha++;
if ( islower( buffer[i] ))
lower++;
else
upper++;
}
else if (isdigit(buffer[i] ))
{
digit++;
}
else if (isspace(buffer[i])){
paces++;
}
else if (ispunct(buffer[i])){
punct++;
}
else
others++;
}
printf("The number of lowercase letters = %d\n", lower);
printf("The number of uppercase letters = %d\n", upper);
printf("The number of spaces = %d\n", spaces);
printf("The number of punctuation characters = %d\n", punct);
printf("The number of alphabets = %d\n", alpha);
printf("The number of digits = %d\n", digit);
}
```

25)Write a function that removes all duplicate spaces from a sentence. Your program should ensure that only one space exists between words

/* We can remove duplicates by checking each character and having a space count. But here we are doing it using the string library function strtok

*/

C

```
#include <stdio.h>


#include <string.h>


int main(){


char input[100];


char output[100];


char temp[100];


char *p;


printf("\nPlease enter a sentence :");


gets(input);


strcpy(temp, input);
```

```
strcpy(output,"");

p = strtok(temp, "");

while ( p != NULL ){

strcat(output,p);

strcat(output,"");

printf("%s\n", p);

p = strtok(NULL, "");

}

printf("%s\n", input);

printf("%s\n", output);

printf("%d\n", strlen(output));

return 0;

}
```

26)Write a function that reads a sentence into a string and splits the sentence into words using library functions

C

```
#include <stdio.h>

#include <string.h>

int main() {

char sentence[100];

char *p;

printf(“\nPlease enter a sentence :”);

gets(sentence);

p = strtok(sentence, “\t “);

while ( p != NULL ){

printf(“%s\n”, p);

p = strtok(NULL, “\t”);

}

return 0;

}
```

27)Write a function that reads a sentence into a string and splits the sentence into words without using library functions

You have to do this without using the strtok function. Since the delimiters are not specified you assume the default delimiters such as spaces, comma, tabs and newlines

You examine character by character and when you encounter a delimiter you have to terminate the word with the NULL character(\0)

```
#include <stdio.h>
#include <string.h>
int main()
{
    char str[100];
    char splitStrings[10][10]; //can store 10 words of 10 characters
    int i,j,cnt;
    printf("Enter a string: ");
    gets(str);
    j=0; cnt=0;
    for(i=0;i<=(strlen(str));i++)
    {
        // if space or NULL found, assign NULL into splitStrings[cnt]
        if(str[i]==' '||str[i]=='\0')
        {
            splitStrings[cnt][j]='\0';
            cnt++;  //for next word
            j=0;    //for next word, init index to 0
        }
        else
        {
            splitStrings[cnt][j]=str[i];
            j++;
        }
   }
    printf("\nOriginal String is: %s",str);
    printf("\nStrings (words) after split by space:\n");
    for(i=0;i < cnt;i++)
        printf("%s\n",splitStrings[i]);
    return 0;
}
```

28)Consider an array of numeric strings where each string is a positive number with anywhere from 1to 10^6 digits. Sort the array's elements in *non-decreasing*, or ascending order of their integer values and print each element of the sorted array on a new line.

Python

```
import sys


n = int(input().strip())
unsorted = []
unsorted_i = 0
for unsorted_i in range(n):
unsorted_t = str(input().strip())
unsorted.append(unsorted_t)
ans=sorted(unsorted, key=lambda x: (len(x),x))
for s in ans:
print(s)
```

29)Write a program to check whether a given number is a happy number or unhappy number.

Happy number: Starting with any positive integer, replace the number by the sum of the squares of its digits, and repeat the process until the number equals 1, or it loops endlessly in a cycle which does not include 1.

Java

```
import java.util.*;

public class Main

{

                public static void main(String[] args)

                {

                                Scanner sc=new Scanner(System.in);

                    int n,s=0,a=0;

        n=sc.nextInt();

                    while(n>9)

                  {

                                while(n!=0)

         {

                                                a=(int)Math.pow(n%10,2);

                                                s=s+a;

                                                    n=n/10;

         }

         n=s; s=0;

     }

     if(n==1)

System.out.println("Happy no.");

    else

System.out.println("Not Happy no.");

                }

}
```