

Activity In LBP:

- 1) Read Problem Statement
- 2) Dividing The problem
- 3) Logic to solve the problem
- 4) Implement the program in C
- 5) Implement the program in Java
- 6) Implement the program in Python
- 7) Run the program for compilation error,
- 8) Submit the program for final approval (1 point) 300+/350

URL Link: <https://www.hackerrank.com/41dec30av2022>

output statements:

C ----->> printf("welcome");

C++ ---->> cout<<"welcome";

Java --->> System.out.print("welcome");

System.out.println("welcome");

System.out.printf("welcome");

Python-->> print("welcome")

Ex: WCP to print "welcome to c lang" on the screen?

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

```
int main(){
```

```
    printf("welcome to c lang");
```

```
}
```

Ex: WCPP to print "welcome to c++" lang on the screen?

```
#include<iostream>

using namespace std;

int main(){

    cout<<"welcome to c++ programming";

}
```

Ex: WJP to print "welcome to java" lang on the screen?

```
public class Demo{

    public static void main(String args[]){

        System.out.println("welcome to java");

    }

}
```

Ex: WPP to print "welcome to Python" lang on the screen?

```
print("welcome to my fav programming:Python")
```

input statements:

C ----->> scanf(formatspecifier+variable_names);

C++ ---->> cin>>var_name;

Java --->> Scanner class and its methods

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

```
Scanner obj = new Scanner(String);
```

```
Scanner obj = new Scanner(File);
```

```
var=obj.nextInt();  
var=obj.nextFloat();  
var=obj.nextDouble();  
var=obj.nextLine();
```

Python -->> input() and typecasting

Ex: WP to read three int values and perform addition operation in C?

```
#include<stdio.h>  
  
int main(){  
    int a,b,c,sum;  
    scanf("%d %d %d",&a,&b,&c);  
    sum=a+b+c;  
    printf("%d",sum);  
}
```

Ex: WP to read three int values and perform addition operation in C++?

```
#include<iostream>  
  
using namespace std;  
  
int main(){  
    int x,y,z,sum;  
    cin>>x>>y>>z;
```

```
sum=x+y+z;  
  
cout<<sum;  
  
}
```

Ex: WP to read three int values and perform addition operation in Java?

```
import java.util.Scanner;  
  
public class Demo{  
  
    public static void main(String args[]){  
  
        Scanner obj = new Scanner(System.in);  
  
        int a = obj.nextInt();  
  
        int b = obj.nextInt();  
  
        int c = obj.nextInt();  
  
        int sum = a+b+c;  
  
        System.out.println(sum);  
  
    }  
  
}
```

Ex: WP to read three int values and perform addition operation in Python?

version1: (beginner level)

```
x=int(input())  
  
y=int(input())  
  
z=int(input())  
  
sum=x+y+z
```

```
print(sum)
```

version2: (expert level)

```
print(sum([int(i) for i in input().split()])))
```

LBP1

Program to check whether the given number is even or odd number

input----> an integer number n

constraint-> $n \geq 0$

output----> even or odd or invalid

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

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

```
    if(n>=0){
```

```
if(n%2==0){  
    printf("even");  
}  
else{  
    printf("odd");  
}  
}  
else{  
    printf("invalid");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n = obj.nextInt();
```

```
if(n>=0){  
    if(n%2==0)  
        System.out.println("even");  
    else  
        System.out.println("odd");  
}  
else  
    System.out.println("invalid");  
}
```

python implementation:

1st version:

```
n=int(input())
```

```
if n>=0:
```

```
    if n%2==0:
```

```
        print("even")
```

```
    else:
```

```
        print("odd")
```

```
else:
```

```
    print("invalid")
```

2nd version:

```
n=int(input())
```

```
print("invalid" if n<0 else ("even" if n%2==0 else "odd"))
```

LBP2

Given an integer n, perform the following conditional actions,

if n is odd, print weird,

if n is even and in the inclusive range of 2 to 5, print Not Weird.

if n is even and in the inclusive range of 6 to 20, print Weird.

if n is even and greater than 20, print Not Weird.

input-----> a number from the user

constraint-> $1 \leq n \leq 100$

output-----> Weird or Not Weird

c implementation:

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

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



```
#include <math.h>

#include <stdlib.h>

int main() {

    int n;

    scanf("%d",&n);

    if(n>=1 && n<=100){

        if(n%2!=0)

            printf("Weird");

        else{

            if(n>=2 && n<=5)

                printf("Not Weird");

            else if(n>=6 && n<=20)

                printf("Weird");

            else

                printf("Not Weird");

        }

    }

    return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        if(n>=1 && n<=100){
```

```
            if(n%2!=0)
```

```
                System.out.println("Weird");
```

```
            else{
```

```
                if(n>=2 && n<=5)
```

```
                    System.out.println("Not Weird");
```

```
                else if(n>=6 && n<=20)
```

```
                    System.out.println("Weird");
```

```
                else
```

```
                    System.out.println("Not Weird");
```

```
            }
```

```
        }
```

```
    }
```

}

python implementation:

```
n=int(input())
```

```
if n>=1 and n<=100:
```

```
    if n%2!=0:
```

```
        print("Weird")
```

```
    else:
```

```
        if n>=2 and n<=5:
```

```
            print("Not Weird")
```

```
        elif n>=6 and n<=20:
```

```
            print("Weird")
```

```
        else:
```

```
            print("Not Weird")
```

LBP3

To check whether the given number is leap year or not.

input-----> year from the user

constraint-> no constraint

output-----> leap year or not leap year

if we take any year it is said to be leap year if it follows the following conditions

1. if it is not a century year and divisible by 4
2. if it is a century year and divisible by 400

<https://www.calendar.best/leap-years.html>

c implementation:

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

```
int main() {
    int year;
    scanf("%d",&year);
    if((year%4==0 && year%100!=0) || (year%400==0)){
        printf("True");
    }
    else{
        printf("False");
    }
    return 0;
}
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int year=obj.nextInt();
```

```
        if((year%4==0 && year%100!=0) || (year%400==0)){
```

```
            System.out.println("True");
```

```
        }
```

```
        else{
```

```
            System.out.println("False");
```

```
        }
```

```
    }
```

```
}
```

python implementation:

1st version:

year=int(input())

if (year%4==0 and year%100!=0) or (year%400==0):

 print("True")

else:

 print("False")

2nd version:

import calendar

print(calendar.isleap(int(input())))

LBP4

The e-commerce company Bookshelf wishes to analyse its monthly sales data between minimum range 30 to max range 100.

The company has categorized these book sales into four groups depending on the number of sales with the help of these groups the company will know which stock they should increase or decrease in their inventory for the next month.

the groups are as follows

sales range

groups

30-50 -----> D

51-60 -----> C

61-80 -----> B

81-100 -----> A

write an alg to find the group for the given book sale count.

input-----> an integer salesCount represent total sales of a book

output-----> a character representing the group of given sale count

constraint---> $30 \leq \text{saleCount} \leq 100$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

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

```
    if(n>=30 && n<=100){
```

```
        if(n>=30 && n<=50)
```

```
printf("D");  
  
else if(n>=51 && n<=60)  
  
    printf("C");  
  
else if(n>=61 && n<=80)  
  
    printf("B");  
  
else  
  
    printf("A");  
  
}  
  
else  
  
    printf("invalid");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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



```
int n=obj.nextInt();

if(n>=30 && n<=100){

    if(n>=30 && n<=50)

        System.out.println("D");

    else if(n>=51 && n<=60)

        System.out.println("C");

    else if(n>=61 && n<=80)

        System.out.println("B");

    else

        System.out.println("A");

}

else

    System.out.println("invalid");

}
```

python implementation:

```
n=int(input())

if n>=30 and n<=100:

    if n>=30 and n<=50:

        print("D")

    elif n>=51 and n<=60:
```

```
print("C")  
  
elif n>=61 and n<=80:  
  
    print("B")  
  
else:  
  
    print("A")  
  
else:  
  
    print("invalid")
```

LBP5

Return the Next Number from the Integer Passed

implement a program that takes a number as an argument, increments the number by +1 and returns the result

input -----> a number from the user

constraints-----> no constraints

output -----> an incremented value

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

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

```
    n++;
```

```
    printf("%d",n);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n =obj.nextInt();
```

```
        System.out.println(++n);
```

```
}
```

```
}
```

python implementation:

```
-----
```

```
print(int(input())+1)
```

LBP6

Free Coffee Cups

For each of the 6 coffee cups I buy, I get a 7th cup free. In total, I get 7 cups.

Implement a program that takes n cups bought and print as an integer the total number of cups I would get.

input -----> n number of cups from user

constraints -----> $n > 0$

output -----> number of cups present have

c implementation:

```
-----
```

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n;  
    scanf("%d",&n);  
    printf("%d",n+n/6);  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        System.out.println(n+n/6);
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())
```

```
print(n+n//6)
```

buy 2 get 1 free

$5+5/2$

$5+2=7$

$2+1$

$2+1$

$1+0$

LBP7

Extract Digits from the number

Implement a program to extract digits from the given number

input -----> a number from the user

constraint -----> $n > 0$

output -----> print digits in line sep by space

c implementation:

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

```
int main() {
    int n,d;
    scanf("%d",&n);
    while(n!=0){
        d=n%10;
        printf("%d ",d);
        n=n/10;
    }
    return 0;
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        while(n!=0){  
            System.out.print((n%10)+" ");  
  
            n=n/10;  
        }  
    }  
}
```

python implementation:

1st version:

```
-----  
  
n=int(input())  
  
while n!=0:  
  
    print(n%10,end=' ')
```


$n = n // 10$

2nd version:

```
for i in input()[::-1]:  
    print(i, end=' ')
```

LBP8

Sum of Digits

Implement a program to calculate sum of digits present in the given number

input -----> a number from the user

constraint -----> $n > 0$

output -----> print sum of digits

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,s=0,d;  
    scanf("%d",&n);  
    while(n!=0){  
        d=n%10;  
        s=s+d;  
        n=n/10;  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
int n=obj.nextInt();

int s=0,d;

while(n!=0){

    d=n%10;

    s=s+d;

    n=n/10;

}

System.out.println(s);

}
```

python implementation:

1st version:

```
n=int(input())
```

```
s=0
```

```
while n!=0:
```

```
    d=n%10
```

```
    s=s+d
```

```
    n=n//10
```

```
print(s)
```

2nd version:

```
print(sum([int(i) for i in list(input())]))
```

LBP9

Sum of even Digits

Implement a program to calculate sum of even digits present in the given number

input -----> a number from the user

constraint -----> $n > 0$

output -----> print sum of even digits

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
int n,d,s=0;

scanf("%d",&n);

while(n!=0){

    d=n%10;

    if(d%2==0)

        s=s+d;

    n=n/10;

}

printf("%d",s);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int d,s=0;
```

```
while(n!=0){  
    d=n%10;  
    if(d%2==0)  
        s=s+d;  
    n=n/10;  
}  
System.out.println(s);  
}
```

python implementation:

```
print(sum([int(i) for i in list(input()) if int(i)%2==0]))
```

LBP10

Sum of odd Digits

Implement a program to calculate sum of odd digits present in the given number

input -----> a number from the user

constraint -----> n>0

output -----> print sum of odd digits

c implementation:

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

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

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int d,s=0;  
  
        while(n!=0){  
            d=n%10;  
  
            if(d%2!=0)  
                s=s+d;  
  
            n=n/10;  
        }  
  
        System.out.println(s);  
    }  
}
```

python implementation:

```
-----  
  
print(sum([int(i) for i in list(input()) if int(i)%2!=0]))
```


LBP11

Sum of prime Digits

Implement a program to calculate sum of prime digits present in the given number

input -----> a number from the user

constraint -----> $n > 0$

output -----> print sum of prime digits

prime ==> prime numbers

prime digits =====> single 0-9

----> prime digits from 0 to 9 =====> 2,3,5,7

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,d,s=0;  
    scanf("%d",&n);  
    while(n!=0){  
        d=n%10;  
        if(d==2 || d==3 || d==5 || d==7)  
            s=s+d;  
        n=n/10;  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
int n=obj.nextInt();

int d,s=0;

while(n!=0){

    d=n%10;

    if(d==2 || d==3 || d==5 || d==7)

        s=s+d;

    n=n/10;

}

System.out.println(s);

}
```

python implementation:

```
print(sum([int(i) for i in list(input()) if i in '2357']))
```

LBP12

Sum of Digits divisible by 3

Implement a program to calculate sum of digits that are divisible by 3 in the given number

input -----> a number from the user

constraint -----> $n > 0$

output -----> print sum of digits that are divisible by 3

from 0 to 9 tell me the digits which are divisible by 3 =====> 3,6,9

```
if(d==3 | |d==6 | |d==9)
```

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d,s=0;
```

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

```
    while(n!=0){
```

```
        d=n%10;
```

```
        if(d==3 | |d==6 | |d==9)
```

```
            s=s+d;
```

```
n=n/10;  
  
}  
  
printf("%d",s);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int d,s=0;  
  
        while(n!=0){  
  
            d=n%10;  
  
            if(d%3==0)  
  
                s=s+d;  
  
            n=n/10;  
  
        }
```

```
System.out.println(s);  
  
}  
  
}
```

python implementation:

```
print(sum([int(i) for i in list(input()) if i in '369']))
```

LBP13

Number of digits

Implement a program to calculate number of digits in the given number

input -----> a number from the user

constraint -----> $n > 0$

output -----> print number of digits in the number

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,c=0;  
    scanf("%d",&n);  
    while(n!=0){  
        c++;  
        n=n/10;  
    }  
    printf("%d",c);  
    return 0;  
}
```

java implementation:

```
-----  
import java.io.*;  
import java.util.*;  
import java.text.*;  
import java.math.*;  
import java.util.regex.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        System.out.println(s.length());  
    }  
}
```

python implementation:

```
-----  
print(len(input()))
```

LBP14

Reverse Integer

Given an integer x, return x with its digits reversed.

input-----> a number from user

constraint ---> $n \geq 0$

output -----> reverse of the given number

c implementation:

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

```
int main() {
    int n,d,r=0;
    scanf("%d",&n);
    while(n!=0){
        d=n%10;
        r=r*10+d;
        n=n/10;
    }
    printf("%d",r);
    return 0;
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),d,r=0;

        while(n!=0){

            d=n%10;

            r=r*10+d;

            n=n/10;

        }

        System.out.println(r);

    }

}
```

python implementation:

```
-----

print(input()[::-1])
```

LBP15

Duck Number

Program to read a number and check whether it is duck number or not.

Hint: A duck number is a number which has zeros present in it,
but no zero present in the beginning of the number.

If any number begin with 0 then it is said to be octal.

input-----> a number from the user

constraint --> $n \geq 0$

output-----> Yes or No

1234 ----> No

1203 ----> Yes

5026 ----> Yes

1000 ----> Yes

4444 ----> No

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d,flag=0;
```

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

```
    while(n!=0){
```

```
        d=n%10;
```

```
        if(d==0){
```

```
            flag=1;
```

```
            break;
```

```
        }
```

```
        n=n/10;
```

```
    }
```

```
    printf((flag==1)?"Yes":"No");
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        System.out.println((obj.nextLine().contains("0"))?"Yes":"No");
    }
}
```

python implementation:

```
print("Yes" if "0" in input() else "No")
```

LBP16

Number of Occurrences

Program to find number of occurrences of the given digit in the number n

input -----> two numbers n and d

constraints-> no constraints

output -----> number of occurrences

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d,key,c=0;
```

```
    scanf("%d %d",&n,&key);
```

```
    while(n!=0){
```

```
        d=n%10;
```

```
        if(d==key){
```

```
            c++;
```

```
        }
```

```
        n=n/10;
```

```
    }
```

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

```
    return 0;
```

```
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),key=obj.nextInt(),c=0,d;  
  
        while(n!=0){  
  
            d=n%10;  
  
            if(d==key){  
  
                c++;  
  
            }  
  
            n=n/10;  
  
        }  
  
        System.out.println(c);  
  
    }  
  
}
```

python implementation:

`print(input().count(input()))`

LBP17

Paliandrome Number

Program to check whether the given number is paliandrome or not

input ----> a number from the user

constraint-> $n > 0$

output ----> Yes or No

c implementation:

`#include <stdio.h>`

`#include <string.h>`

`#include <math.h>`

`#include <stdlib.h>`

`int main() {`

`int n,r,temp,d;`


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

r=0;

temp=n;

while(n!=0){

    d=n%10;

    r=r*10+d;

    n=n/10;

}

printf((temp==r)?"Yes":"No");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),r,temp,d;
```

```
r=0;

temp=n;

while(n!=0){

    d=n%10;

    r=r*10+d;

    n=n/10;

}

System.out.println((temp==r)?"Yes":"No");

}

}
```

python implementation:

```
s=input()

print("Yes" if s==s[::-1] else "No")
```

LBP18

Check Birth Day

Lisa always forgets her birthday which is on th 5th July.

So develop a function/method which will be helpful to remember her birthday.

The function/method checkBirthday return an integer 1, if it is her birthday else return 0.

the function/method checkBirthday accepts two arguments.

Month, a string representing the month of her birth and day, an integer representing the data of her birthday.

input -----> month & day

constraints -----> no

output -----> 1 or 0

c implementation:

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

```
int main() {
    int day;
    char month[10];
    scanf("%s",month);
    scanf("%d",&day);
    if(strcmp(month,"july")==0 && day==5)
        printf("1");
}
```

```
else  
  
    printf("0");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int day=obj.nextInt();  
  
        if(s.equals("july") && day==5)  
  
            System.out.println(1);  
  
        else  
  
            System.out.println(0);  
  
    }  
  
}
```

2nd version:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static int checkBirthday(String month,int day){
```

```
        //write your logic here
```

```
        return (month.equals("july")&&day==5)?1:0;
```

```
    }
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        int day=obj.nextInt();
```

```
        System.out.println(checkBirthday(s,day));
```

```
    }
```

```
}
```

python implementation:

```
month=input()
```

```
day=input()
if month=="july" and day=='5':
    print(1)
else:
    print(0)
```

LBP19

Decimal to Binary

A network protocol specifies how data is exchanged via transmission media.

The protocol converts each message into a stream of 1's and 0's.

Given a decimal number, write an algorithm to convert the number into a binary form.

input -----> a number
constraint ----> $n \geq 0$
output -----> binary number

c implementation:

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[10],i,d;  
    scanf("%d",&n);  
    i=0;  
    while(n!=0){  
        d=n%2;  
        a[i++]=d;  
        n=n/2;  
    }  
    for(i=i-1;i>=0;i--){  
        printf("%d",a[i]);  
    }  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    System.out.println(Integer.toBinaryString(obj.nextInt()));  
}  
}
```

python implementation:

```
-----  
print(bin(int(input()))[2:])
```

LBP20

Lucky Customer

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 n is the sum of the preceding number.

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 one who has placed that order. Write an algorithm to help the website find the lucky customer.

input -----> a number

constraint ---> $n > 0$

output -----> a number

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int fib(int n){
```

```
    if(n==0 | n==1)
```

```
        return n;
```

```
    else
```

```
        return fib(n-1)+fib(n-2);
```

```
}
```

```
int main() {
```

```
    int n;
```

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

```
    printf("%d",fib(n));
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static int fib(int n){
```

```
        if(n==0 || n==1)
```

```
            return n;
```

```
        else
```

```
            return fib(n-1)+fib(n-2);
```

```
    }
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        System.out.println(fib(n));
```

```
    }
```

```
}
```

python implementation:

```
def fib(n):
```

```
if n==0 or n==1:  
    return n  
else:  
    return fib(n-1)+fib(n-2)  
print(fib(int(input())))
```

LBP21

Christmas offer

An e-commerce company plans to give their customers a special discount for the Christmas,

they are planning to offer a flat discount.

The discount value is calculated as the sum of all prime digits in the total bill amount.

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

input ----> the input consists of an integer order value representing the total bill amount

condition-> no conditions

output ---> print an integer representing discount value for the given total bill amount.

12500.00 =====> 2+5=7%

12540.00 =====> 2+5=7%

sum of prime digits

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d,s=0;
```

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

```
    while(n!=0){
```

```
        d=n%10;
```

```
        if(d==2 || d==3 || d==5 || d==7)
```

```
            s=s+d;
```

```
        n=n/10;
```

```
    }
```

```
    printf("%d",s);
```

```
return 0;

}

java implementation:

-----

import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int d,s=0;

        while(n!=0){

            d=n%10;

            if(d==2 || d==3 || d==5 || d==7)

                s=s+d;

            n=n/10;

        }

        System.out.println(s);

    }

}
```

python implementation:

```
print(sum([int(i) for i in list(input()) if i in '2357']))
```

LBP22

Niven Number

Write a program to accept a number and check and display whether it is a Niven Number or not.

Niven Number is that a number which is divisible by its sum of digits.

input ----> a number

constraint-> $n > 0$

output ----> Niven Number or Not

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,temp,s,d;  
    scanf("%d",&n);  
    temp=n;  
    s=0;  
    while(n!=0)  
    {  
        d=n%10;  
        s=s+d;  
        n=n/10;  
    }  
    if(temp%s==0)  
        printf("Yes");  
    else  
        printf("No");  
    return 0;  
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int temp,s,d;

        temp=n;

        s=0;

        while(n!=0)

        {

            d=n%10;

            s=s+d;

            n=n/10;

        }

        System.out.println((temp%s==0)?"Yes":"No");

    }

}
```

python implementation:

```
n=int(input())  
temp=n  
s=0  
while n!=0:  
    d=n%10  
    s=s+d  
    n=n//10  
print("Yes" if temp==s else "No")
```

LBP23

A Special two digit number

A special two digit number is a number such that when the sum of its digits is added to the product of its digits, the result should be equal to the original two-digit number.

Implement a program to accept a two digit number and check whether it is a special two digit number or not.

input -----> a two digit number

constraint-> $10 \leq n \leq 99$

output ----> special two digit number or not

12 ==> (1+2)+(1*2) =3+2=5 No

59 ==> (5+9)+(5*9) =14+45=59 Yes

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a,b,c;
```

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

```
    a=n%10;
```

```
    b=(n/10)%10;
```

```
    c=(a+b)+(a*b);
```

```
    printf((c==n)?"Yes":"No");
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),a,b,c;

        a=n%10;

        b=(n/10)%10;

        c=(a+b)+(a*b);

        System.out.println((c==n)?"Yes":"No");

    }

}
```

python implementation:

```
-----

n=int(input())

a=n%10

b=(n//10)%10

c=(a+b)+(a*b)

print("Yes" if c==n else "No")
```

LBP24

Sum of even numbers

Implement a program to find sum of even number between x and y both are inclusive.

input -----> two int values

constraint-> no

output ----> sum of even numbers between x and y

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int x,y,i,s;
```

```
    scanf("%d %d",&x,&y);
```

```
    s=0;
```

```
for(i=x;i<=y;i++)  
{  
    if(i%2==0)  
        s=s+i;  
}  
printf("%d",s);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int x=obj.nextInt(),y=obj.nextInt(),i,s;  
        s=0;  
        for(i=x;i<=y;i++)  
        {
```

```
        if(i%2==0)

            s=s+i;

        }

        System.out.println(s);

    }

}
```

python implementation:

```
x=int(input())
y=int(input())
s=0
for i in range(x,y+1):
    if i%2==0:
        s=s+i
print(s)
```

range(10,20) -----> 10,11,12,13,14,15,16,17,18,19

range(10,20+1) ----> 10,11,12,13,14,15,16,17,18,19,20

range(a,b) -----> a to b-1

range(a,b+1) -----> a to b

LBP25

Celsius to Fahrenheit

Create a function/method to convert celsius to fahrenheit.

input -----> celsius

constraint --> no

output -----> Fahrenheit

formula: $F = (C * 9/5) + 32$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int c;
```

```
    scanf("%d",&c);
```

```
    printf("%d",(c*9/5)+32);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        System.out.println((obj.nextInt()*9/5)+32);
```

```
    }
```

```
}
```

python implementation:

```
print((int(input())*9//5)+32)
```

LBP26

Fahrenheit to Celsius

Program to convert fahrenheit to celsius.

input -----> fahrenheit

constraint --> no

output -----> celsius

formula: $C = (F-32)*5/9$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int F;
```

```
    scanf("%d",&F);
```

```
    printf("%d",(F-32)*5/9);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        System.out.println((obj.nextInt()-32)*5/9);
```

```
    }
```

```
}
```

python implementation:

```
#print((int(input())-32)*(5//9))
```

```
print(int(((int(input())-32)*5)/9))
```

LBP27

Find The Sequence Sum

Given three integers i,j&k, a sequence sum to be the value of

$i+(i+1)+(i+2) \dots +j+(j-1)+(j-2) \dots +k$

(increment from i until it equals to j,

then decrement from j until equals k).

Given values i,j,k.

caluclate the sequence sum as described.

```
int getSequenceSum(int,int,int);
```

input -----> Three int values

constraints--> no

output -----> sum basd on given constraints

5,9,6

$5+6+7+8+9+8+7+6=56$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
int i,j,k,s=0;

scanf("%d %d %d",&i,&j,&k);

while(i<=j){s=s+(i++);}

while(j!=k){s=s+(--j);}

printf("%d",s);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int i,j,k,s;

        i=obj.nextInt();

        j=obj.nextInt();

        k=obj.nextInt();

        s=0;
```

```
while(i<=j){  
    s=s+(i++);  
}  
while(j!=k){  
    s=s+(-j);  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
i=int(input())  
j=int(input())  
k=int(input())  
s=0  
while i<=j:  
    s=s+i  
    i=i+1  
while j!=k:  
    j=j-1  
    s=s+j  
print(s)
```

LBP28

You are climbing a stair case. It takes n steps to reach to the top.

Each time you can either climb 1 or 2 steps.

In how many distinct ways can you climb to the top?

Note: Given n will be a positive integer.

input -----> a number from the user

constraints --> no

output -----> number of ways

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int fib(int n)
```

```
{
```

```
    if(n==0 | n==1)
```

```
        return 1;

    else

        return fib(n-1)+fib(n-2);

}

int main() {

    int n;

    scanf("%d",&n);

    printf("%d",fib(n));

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    static int fib(int n)

    {

        if(n==0 || n==1)

            return 1;

        else
```

```
        return fib(n-1)+fib(n-2);  
    }  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        System.out.println(fib(obj.nextInt()));  
    }  
}
```

python implementation:

```
-----  
  
def fib(n):  
    if n==0 or n==1:  
        return 1  
    else:  
        return fib(n-1)+fib(n-2)  
  
print(fib(int(input())))
```

LBP29

Prime Number or Not

Write a program to check whether the given number is prime number or not.

A number is said to prime if it is having only two factors. i.e. 1 and number itself.

input -----> a number from the use

constraint--> $n > 1$

output -----> true or false

c implementation:

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

```
int main() {
    int n,i,f=0;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
            f++;
    }
    printf((f==2)?"true":"false");
}
```

```
return 0;  
  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),i,f=0;  
        for(i=1;i<=n;i++)  
        {  
            if(n%i==0)  
                f++;  
        }  
        System.out.println(f==2);  
    }  
}
```

python implementation:

```
-----  
n=int(input())  
  
f=0  
  
for i in range(1,n+1):  
    if n%i==0:  
        f=f+1  
  
print(str(f==2).lower())
```

LBP30

Valid Palindrome

Given a string,

determine if it is a Palindrome string or not.

A String is Palindrome if it is equal to reverse of the original string.

input -----> A String from the user

constraint--> Non-empty String

output -----> Palindrome or not

"hai", "iah" -----> invalid

"liril", "liril" ---> valid

c-----> core concept

java & py ----> predefined

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int low,high,ispali=1;
```

```
    char s[100];
```

```
    scanf("%s",s);
```

```
    low=0;
```

```
    high=strlen(s)-1;
```

```
    while(low<=high)
```

```
    {
```

```
        if(s[low]!=s[high])
```

```
        {
```

```
            ispali=0;
```

```
            break;
```

```
        }
```

```
low=low+1;

high=high-1;

}

printf((ispali==1)?"valid":"invalid");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

        StringBuffer sb = new StringBuffer(s);

        sb.reverse();

        System.out.println((s.equals(sb.toString()))?"valid":"invalid");

    }

}
```

python implementation:

```
s=input()
```

```
print("valid" if s==s[::-1] else "invalid")
```

200 Screen Shots of Programs

R&D

program -----> change statements -----> new program

Repeated every time

LBP31

Create PIN using Three given numbers

"Secure Assets Private Ltd", a small company that deals with lockers has recently started manufacturing digital locks which can be locked and unlocked using PINs (passwords).

You have been asked to work on the module that is expected to generate PINs using three input numbers.

The three given input numbers will always consist of three digits each

i.e. each of them will be in the range ≥ 100 and ≤ 999 .

Bellow are the rules for generating the PIN.

1. The PIN should made up of 4 digits.
2. The unit (ones) position of the PIN should be the least of the units position of the three numbers.
3. The tens position of the PIN should be the least of the tens position of the three input numbers.
4. The hundreds position of the PIN should be least of the hundreds position of the three numbers.
5. The thousands position of the PIN should be the max of all digits in the three input numbers.

input -----> three numbers

constraints ----> all the numbers must be in the range of ≥ 100 and ≤ 999

output -----> PIN value

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int min(int a,int b,int c)
{
    return (a<b && a<c)?a:(b<c?b:c);
}

int max(int a,int b,int c)
{
    return (a>b && a>c)?a:(b>c?b:c);
}

int maxDigit(int n)
{
    int m=0,d;
    while(n!=0)
    {
        d=n%10;
        if(d>m)
            m=d;
        n=n/10;
    }
    return m;
}

int main() {
    int n1,n2,n3;

    scanf("%d %d %d",&n1,&n2,&n3);
```



```
int w = min(n1%10,n2%10,n3%10);

int x = min((n1/10)%10,(n2/10)%10,(n3/10)%10);

int y = min((n1/100)%10,(n2/100)%10,(n3/100)%10);

int z = max(maxDigit(n1),maxDigit(n2),maxDigit(n3));

printf("%d",z*1000+y*100+x*10+w);

return 0;

}
```

java implementation:

```
-----

import java.io.*;

import java.util.*;

public class Solution {

    static int maxDigit(int n)

    {

        int m=0,d;

        while(n!=0)

        {

            d=n%10;

            if(d>m)

                m=d;

            n=n/10;

        }

    }

}
```

```
}

return m;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n1=obj.nextInt();

    int n2=obj.nextInt();

    int n3=obj.nextInt();

    int w = Math.min(Math.min(n1%10,n2%10),n3%10);

    int x = Math.min(Math.min((n1/10)%10,(n2/10)%10),(n3/10)%10);

    int y = Math.min(Math.min((n1/100)%10,(n2/100)%10),(n3/100)%10);

    int z = Math.max(Math.max(maxDigit(n1),maxDigit(n2)),maxDigit(n3));

    System.out.println(z*1000+y*100+x*10+w);

}

}
```

python implementation:

```
-----

n1=[int(i) for i in input()]
n2=[int(i) for i in input()]
n3=[int(i) for i in input()]
w=min(n1[2],n2[2],n3[2])
x=min(n1[1],n2[1],n3[1])
```

```
y=min(n1[0],n2[0],n3[0])  
z=max(max(n1),max(n2),max(n3))  
print(z*1000+y*100+x*10+w)
```

LBP32

Program to count number of special characters and white spaces in a given string.

input -----> A string from the user

constraint ---> non-empty string

output -----> number of special characters

a-z

A-Z

0-9

if(()) or () or ()

continue

else

C++

print c

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int c=0,i;
```

```
    //scanf("%s",s); //read char and stop reading when it reaches space
```

```
    scanf("%[^\n]s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        if((s[i]>='a'&& s[i]<='z') || (s[i]>='A'&& s[i]<='Z') || (s[i]>='0'&& s[i]<='9'))
```

```
            continue;
```

```
        else
```

```
            c++;
```

```
    }
```

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

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        int i,c=0;
```

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

```
        {
```

```
            char ch = s.charAt(i);
```

```
            if((ch>='a' && ch<='z') || (ch>='A' && ch<='Z') || (ch>='0' && ch<='9'))
```

```
                continue;
```

```
            else
```

```
                c++;
```

```
        }
```

```
        System.out.println(c);
```

```
}
```

```
}
```

python implementation:

```
-----
```

```
s=input()
```

```
c=0
```

```
for i in s:
```

```
    if not i.isalnum():
```

```
        c=c+1
```

```
print(c)
```

LBP33

An e-commerce company plans to give their customers a discount for the newyears 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 in the customers total bill amount.

if no odd digits is present in the bill amount, then discount will be zero.

input -----> the input consists of an integer bill amount, representing the customers total bill amount.

output -----> print an integer representing the discount for the given total bill amount.

constraint ---> $n > 0$

sum of odd digits

c implementation:

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

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

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int d,s=0;  
  
        while(n!=0){  
            d=n%10;  
  
            if(d%2!=0)  
                s=s+d;  
  
            n=n/10;  
        }  
  
        System.out.println(s);  
    }  
}
```

python implementation:

```
print(sum([int(i) for i in list(input()) if int(i)%2!=0]))
```

LPB34

Email name should be starts with alphabet and should follw by number or underscore.

It should contains either numbers or underscore finally ends with @gmail.com only,

Then given email id is true otherwise false.

input -----> email id

constraint -> lowercase alphabet [a-z] followed by underscore or digit and gmail.com

output -----> true or false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
#define isAlpha(ch) (ch>='a' && ch<='z')
```

```
#define isDigit(ch) (ch>='0' && ch<='9')
```

```
int isEmail(char s[])
{
    int i;
    for(i=0;s[i]&&isAlpha(s[i]);i++);
    if((i==0 && isDigit(s[i])) || (i==0 && s[i]=='_'))
        return 0;
    else if(isDigit(s[i]))//only for abc1@gmail.com , digit
    {
        i++;
        if(s[i]!='_' && strcmp(s+i,"@gmail.com")==0)
            return 1;
        else
            return 0;
    }
    else if(s[i]=='_' && strcmp(s+1+i,"@gmail.com")==0)
    {
        return 1;
    }
    else
    {
        return 0;
    }
}
```

```
int main() {  
    char s[100];  
    scanf("%s",s);  
    if(isEmail(s))  
        printf("true");  
    else  
        printf("false");  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
import java.util.regex.*;  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        Pattern p = Pattern.compile("[a-z]+[0-9|_]+@gmail[.]com");  
        Matcher m = p.matcher(s);
```

```
if(m.find() && (m.group()).equals(s))  
    System.out.println("true");  
else  
    System.out.println("false");  
}  
}
```

python implementation:

```
-----  
import re  
  
m = re.fullmatch("[a-z]+[0-9|_]+@gmail[.].com", input())  
print("true" if m != None else "false")
```

LBP35

The IT company "Soft ComInfo" has decided to transfer its messages through the N/W using

new encryption technique.

The company has decided to encrypt the data using the non-prime number concept.

The message is in the form of a number and the sum of non-prime digits present in the message is used as the encryption key.

Write an algorithm to determine the encryption key.

input -----> The input consists of an integer numMsg representing the numeric form of the message.

output -----> print an integer representing the encryption key.

note: Digit 1 and 0 are considered as a prime number.

c implementation:

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

```
int main() {
    int n,d,s;
    scanf("%d",&n);
    s=0;
    while(n!=0)
    {
```

```
d=n%10;

if(d==4 || d==6 || d==8 || d==9)

    s=s+d;

n=n/10;

}

printf("%d",s);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int d,s=0;

        while(n!=0)

        {

            d=n%10;
```

```
if(d==4 || d==6 || d==8 || d==9)

    s=s+d;

    n=n/10;

}

System.out.println(s);

}

}
```

python implementation:

```
print(sum([int(i) for i in input() if i in "4689"]))
```

LBP36

Implement a program to return First capital letter in a String

input -----> A string from the user

constraint --> non-empty string

output -----> First Capital letter

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

#include <ctype.h>

int main() {

    char s[100];

    scanf("%s",s);

    for(int i=0;s[i];i++)

    {

        //if(s[i]>='A' && s[i]<='Z')

        if(isupper(s[i]))

        {

            printf("%c",s[i]);

            break;

        }

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

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

        {

            if(Character.isUpperCase(s.charAt(i)))

            {

                System.out.println(s.charAt(i));

                break;

            }

        }

    }

}
```

python implementation:

```
s=input()
```

```
for i in s:
```

```
if i.isupper():
```

```
    print(i)
```

```
    break
```

LBP37

Implement a program to calculate toggle case of each characters of a string

input -----> A String from user

constraint --> non-empty String

output -----> toggle case string

if A->a or a->A

$A = 65 + 32 = 97 = a$

$a = 97 - 32 = 65 = A$

+32 in C and Java

swapcase()

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100];

    scanf("%s",s);

    for(int i=0;s[i];i++)

    {

        if(s[i]>='A' && s[i]<='Z')

            printf("%c",s[i]+32);

        else if(s[i]>='a' && s[i]<='z')

            printf("%c",s[i]-32);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s=obj.nextLine();  
        for(int i=0;i<s.length();i++)  
        {  
            if(s.charAt(i)>='A' && s.charAt(i)<='Z')  
                System.out.print((char)(s.charAt(i)+32));  
            else if(s.charAt(i)>='a' && s.charAt(i)<='z')  
                System.out.print((char)(s.charAt(i)-32));  
        }  
    }  
}
```

python implmenetation:

```
print(input().swapcase())
```

LBP38

A company launched a new text editor that allows users to enter english letters, numbers and white spaces only.

If a user attempts to enter any other type of characters, it is counted as miss.

Given a String text,

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

input -----> String

constraint ---> non-empty string

output -----> number of misses

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,c=0;
```

```
    scanf("%[^\\n]s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
if((s[i]>='a'&& s[i]<='z') || (s[i]>='A'&& s[i]<='Z') || (s[i]>='0'&& s[i]<='9') || (s[i]==' '))
    continue;
else
    c++;
}
printf("%d",c);
return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        int c=0;
```

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

```
{  
    char ch=s.charAt(i);  
    if((ch>='a'&&ch<='z') || (ch>='A'&&ch<='Z') || (ch>='0'&&ch<='9') || (ch=='  
''))  
        continue;  
    else  
        c++;  
}  
System.out.println(c);  
}  
}
```

python implementation:

```
-----  
s=input()  
c=0  
for i in s:  
    if i.isalnum() or i.isspace():  
        continue  
    else:  
        c=c+1  
print(c)
```

[^a-zA-Z0-9]

LBP39

Implement the following function

```
int BlackJack(int n1,int n2);
```

the function accepts two +ve integers n1 and n2 as its arguments.

Implement the function on given two vlaues to return an int value as follows

return whichever value is nearest to 21 without going over. Return if they go both go over.

input -----> two int values n1 and n2

constraint ---> no

output -----> 0 or n1 or n2

$n1 > 21$ and $n2 > 21 \implies 0$

$n1 > 21 \implies n2$

$n2 > 21 \implies n1$

$n1 < 21$ and $n2 < 21 \implies n1$ or $n2$ max

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
int BJ(int n1,int n2)  
{  
    if(n1>21 && n2>21)  
        return 0;  
    if(n1>21)  
        return n2;  
    else if(n2>21)  
        return n1;  
    else  
        return (n1>n2)?n1:n2;  
}  
  
int main() {  
    int n1,n2;  
  
    scanf("%d %d",&n1,&n2);  
  
    printf("%d",BJ(n1,n2));  
  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static int BJ(int n1,int n2)
```

```
    {
```

```
        if(n1>21 && n2>21)
```

```
            return 0;
```

```
        else if(n1>21)
```

```
            return n2;
```

```
        else if(n2>21)
```

```
            return n1;
```

```
        else
```

```
            return Math.max(n1,n2);
```

```
    }
```

```
    public static void main(String[] args) {
```

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

```
        int n1=obj.nextInt();
```

```
        int n2=obj.nextInt();
```

```
        System.out.println(BJ(n1,n2));
```

```
}  
  
}
```

python implementation:

```
-----  
  
def BJ(n1,n2):  
    if n1>21 and n2>21:  
        return 0  
    elif n1>21:  
        return n2  
    elif n2>21:  
        return n1  
    else:  
        return max((n1,n2))  
  
n1=int(input())  
n2=int(input())  
print(BJ(n1,n2))
```

LBP40

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 during transmission,

they plan to reverse the data first.

Write an algorithm to reverse the data first

input ----> an integer data, representing the data to be transmitted

output ----> print an integer representing the given data in reverse form

c implementation:

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

```
int main() {
    int n,d,r=0;
    scanf("%d",&n);
    while(n!=0){
        d=n%10;
        r=r*10+d;
        n=n/10;
    }
```

```
printf("%d",r);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),d,r=0;  
  
        while(n!=0){  
  
            d=n%10;  
  
            r=r*10+d;  
  
            n=n/10;  
  
        }  
  
        System.out.println(r);  
  
    }  
  
}
```

python implementation:

```
-----  
  
print(input()[::-1])
```

LBP41

One Time Password

A company wishes to devise an order confirmation procedure.

They plan to require an extra confirmation instead of simply auto-confirming the order at the time it is placed.

for this purpose, the system will generate one time password to be shared with the customer.

The customer who is placing the order has to enter the OTP to confirm the order.

The OTP generated for the requested order ID, as the product of the digits in orderID.

Write an algorithm to find the OTP for the OrderID.

input ----> an integer representing order id

output ----> an integer representing OTP

p=1;

```
while(n!=0)
```

```
{
```

```
    d=n%10;
```

```
    p=p*d;
```

```
    n=n/10;
```

```
}
```

```
print p
```

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,p,d;
```

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

```
    p=1;
```

```
    while(n!=0)
```

```
{
```

```
    d=n%10;
```

```
    p=p*d;
```

```
n=n/10;  
  
}  
  
printf("%d",p);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),p=1,d;  
        while(n!=0){  
            d=n%10;  
            p=p*d;  
            n=n/10;  
        }  
        System.out.println(p);  
    }  
}
```



```
}
```

python implementation:

```
import math
```

```
print(math.prod([int(i) for i in input()])))
```

python 3.8 version

LBP42

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

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

input -----> two space sepearted integers RL and RR.

output ----> sum of the prime numbers between RL and RR.

x and y

s=0;

```
for(i=x;i<=y;i++)
```

```
{
```

```
    if(isprime(i))
```

```
        s=s+i;
```

```
}
```

```
print s
```

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int isprime(int n)
```

```
{
```

```
    int f=0,i;
```

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

```
    {
```

```
        if(n%i==0)
```

```
            f++;
```

```
    }
```

```
    return f==2;//1 or 0
```

```
}
```

```
int main() {  
    int n1,n2,i,s=0;  
    scanf("%d %d",&n1,&n2);  
    for(i=n1;i<=n2;i++)  
    {  
        if(isprime(i))  
            s=s+i;  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
-----  
import java.io.*;  
import java.util.*;  
  
public class Solution {  
    static boolean isprime(int n)  
    {  
        int f=0,i;  
        for(i=1;i<=n;i++)
```

```
{  
    if(n%i==0)  
        f++;  
}  
return f==2;  
}  
  
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n1=obj.nextInt(),n2=obj.nextInt(),s=0,i;  
    for(i=n1;i<=n2;i++)  
    {  
        if(isprime(i))  
            s=s+i;  
    }  
    System.out.println(s);  
}  
}
```

python implementation:

```
def isprime(n):  
    f=0  
  
    for i in range(1,n+1):
```

```
if n%i==0:
    f=f+1
return f==2
n1=int(input())
n2=int(input())
s=0
for i in range(n1,n2+1):
    if isprime(i):
        s=s+i
print(s)
```

LBP43

An e-Commerce company plans to give thier customers a discount for the newyears holiday. The discount will be calcualted on the basis of the bill amount of the order place. The discount amount is the productof the sum of all odd digits and the sum of all even digits of the customers total bill amount.

input ----> an integer bill amount, representing the total bill amount of the customer.

output ---> print an inteer representing the discount amount for the given total bill.

se=0

```
so=0
while(n!=0)
{
    d=n%10;
    if(d%2==0)
        se=se+d;
    else
        so=so+d;
    n=n/10;
}
print se*so
```

c implementation:

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

```
int main() {
    int n,se=0,so=0,d;
    scanf("%d",&n);
    while(n!=0)
```

```
{  
  
    d=n%10;  
  
    if(d%2==0)  
        se=se+d;  
  
    else  
        so=so+d;  
  
    n=n/10;  
  
}  
  
printf("%d",se*so);  
  
return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
  
public class Solution {  
  
  
  
  
  
  
  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),se=0,so=0,d;
```

```
while(n!=0)
{
    d=n%10;
    if(d%2==0)
        se=se+d;
    else
        so=so+d;
    n=n/10;
}
System.out.println(se*so);
}
```

python implementation:

```
n=int(input())
se=0
so=0
while n!=0:
    d=n%10
    if d%2==0:
        se=se+d
    else:
```



```
so=so+d
```

```
n=n//10
```

```
print(se*so)
```

LBP44

War of Numbers

There is a great war between the even and odd numbers.

Many numbers already lost their life in this war and it's your task to end this.

You have to determine which group sums larger. the even, or the odd, the larger group wins.

Create a function that takes an array of integers, sums the even and odd numbers separately,

then return the difference between sum of even and odd numbers.

input -----> number and array elements

constraint ---> no

output -----> difference between sum of even and odd numbers

```
se=0
```

```
so=0
```

```
for(i=0;i<n;i++)  
{  
    if(a[i]%2==0)  
        se=se+a[i]  
    else  
        so=so+a[i]  
}  
diff=abs(se-so)
```

c implementation:

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

```
int main() {  
    int n,a[100],i,se=0,so=0;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
    {  
        scanf("%d",&a[i]);  
    }
```

```
for(i=0;i<n;i++)  
{  
    if(a[i]%2==0)  
        se=se+a[i];  
    else  
        so=so+a[i];  
}  
printf("%d",abs(se-so));  
return 0;  
}
```

```
int sum(int a[]){  
    for(i=0;i<n;i++)  
    {  
        if(a[i]%2==0)  
            se=se+a[i];  
        else  
            so=so+a[i];  
    }  
    return abs(se-so);  
}
```

```
printf("%d",sum(a));
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),se=0,so=0,i;
```

```
        int a[] = new int[n];
```

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

```
            a[i]=obj.nextInt();
```

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

```
        {
```

```
            if(a[i]%2==0)
```

```
                se=se+a[i];
```

```
            else
```

```
                so=so+a[i];
```

```
        }
```

```
        System.out.println(Math.abs(se-so));
```

```
}
```

```
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
se=0
```

```
so=0
```

```
for i in L:
```

```
    if i%2==0:
```

```
        se=se+i
```

```
    else:
```

```
        so=so+i
```

```
print(abs(se-so))
```

Rakesh Code:

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

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

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

```
#include <stdlib.h>
```

```
int diff(int []);
```

```
int main() {  
    int n,arr[20],i,res;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&arr[i]);  
    res=diff(arr,n);  
    if(res<0){  
        printf("%d",abs(res));  
    }  
    else  
        printf("%d",res);  
    return 0;  
}
```

```
int diff(int arr[],int n){  
    int i,sumeven=0,sumodd=0,res;  
    for (i=0;i<n;i++){  
        if(arr[i]%2==0)  
            sumeven=sumeven+arr[i];  
        else
```

```
        sumodd=sumodd+arr[i];  
    }  
  
    res=sumeven-sumodd;  
  
    return res;  
}
```

LBP45

Perfect Number

Create a function that tests whether or not an integer is a perfect number.

A perfect number is a number that can be written as sum of its factors.

(equal to sum of its proper divisors) excluding the number itself.

input -----> a number from the user

constraint -> $n > 0$

output -----> true or false

$4 \implies 1, 2, 4 \implies 1, 2 \implies 1+2=3$

$6 \implies 1, 2, 3, 6 \implies 1, 2, 3 \implies 1+2+3=6$

s=0;

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

```
{  
    if(n%i==0)  
        s=s+i;  
}  
if(s==n) print yes else no
```

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
    int n,s=0,i;  
    scanf("%d",&n);  
    for(i=1;i<n;i++)  
    {  
        if(n%i==0)  
            s=s+i;  
    }  
    printf((s==n)?"true":"false");  
    return 0;
```



```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),s=0,i;
```

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

```
        {
```

```
            if(n%i==0)
```

```
                s=s+i;
```

```
        }
```

```
        System.out.println(s==n);
```

```
    }
```

```
}
```

python implementation:

```
-----  
n=int(input())  
  
s=0  
  
for i in range(1,n):  
    if n%i==0:  
        s=s+i  
  
print("true" if n==s else "false")
```

LBP46

Magic Date

Program to read date, month and year from the user and check whether it is magic date or not.

Here are the rules for magic date.

- 1) mm*dd is a 1-digit number that matches the last digit in YYYY
- 2) mm*dd is a 2-digit number that matches the last two digits in YYYY
- 3) mm*dd is a 3-digit number that matches the last three digits in YYYY

input -----> three int values

constraint----> no

output -----> true or false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int d,m,y;
```

```
    scanf("%d-%d-%d",&d,&m,&y);
```

```
    if(d*m==y%10 || d*m==y%100 || d*m==y%100)
```

```
        printf("true");
```

```
    else
```

```
        printf("false");
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String[] s = obj.nextLine().split("-");//5,1,2010
```

```
        System.out.println(s[2].endsWith(Integer.toString(Integer.parseInt(s[0])*Integer.parseInt(s[1]))));
```

```
    }
```

```
}
```

python implementation:

```
L=[i for i in input().split("-")]
```

```
print(str(L[2].endswith(str(int(L[0])*int(L[1])))).lower())
```

array elements in java are separated by - or , ?

ans: space

LBP47

Oddish or Evenish

Create a function that determines whether a number is Oddish or Evenish.

A number is Oddish if the sum of all of its digits is Odd,

and number is Evenish if the sum of all of its digits is even,

if a number is Oddish return Oddish else return Evenish.

input -----> a number

constraint -----> $n > 0$

output -----> Oddish or Evenish

sum of digits % 2 == 0 then Evenish else Oddish

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
int n,s=0,d;

scanf("%d",&n);

while(n!=0)

{

    d=n%10;

    s=s+d;

    n=n/10;

}

printf((s%2==0)?"Evenish":"Oddish");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),s=0;

        while(n!=0)
```

```
{  
    s=s+(n%10);  
    n=n/10;  
}  
System.out.println((s%2==0?"Evenish":"Oddish");  
}  
}
```

python implementation:

```
print("Evenish" if sum([int(i) for i in input()])%2==0 else "Oddish")
```

LBP48

Accept video length in minutes the format is mm:ss in String format,
create a function that takes video length and return it in seconds.

input -----> video length in mm:ss

constraint----> no

output -----> length in seconds

01234

01:00 ==> 60

02:05 ==> 125

22:01 ==> 1203

$2*10+2$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int convert(char s) //a=97,A=65,0=48 (48,49,50,...)
```

```
{
```

```
    return s-48;
```

```
}
```

```
int main() {
```

```
    char s[100];
```

```
    int n1,n2;
```

```
    scanf("%s",s);
```

```
    if(s[0]=='0')
```

```
        n1=convert(s[1]);
```

```
    else
```

```
        n1=convert(s[0])*10+convert(s[1]);
```



```
if(s[3]!='0')  
    n2=convert(s[4]);  
else  
    n2=convert(s[3])*10+convert(s[4]);  
printf("%d",n1*60+n2);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();//ab:cd  
        int n1,n2;  
        if(s.charAt(0)=='0')  
            n1=Integer.valueOf(s.charAt(1)-48);  
        else  
            n1=Integer.valueOf(s.charAt(0)-48)*10+Integer.valueOf(s.charAt(1)-48);  
    }  
}
```

```
if(s.charAt(3)=='0')  
    n2=Integer.valueOf(s.charAt(4)-48);  
else  
    n2=Integer.valueOf(s.charAt(3)-48)*10+Integer.valueOf(s.charAt(4)-48);  
System.out.println(n1*60+n2);  
}  
}
```

2nd version:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();//ab:cd
```

```
        //String s[]=obj.nextLine().split(":");//s[0],s[1]
```

```
        int n1,n2;
```

```
        if(s.charAt(0)=='0')
```

```
            n1=Integer.parseInt(s.substring(1,2));
```

```
        else
```

```
n1=Integer.parseInt(s.substring(0,2));  
if(s.charAt(3)=='0')  
    n2=Integer.parseInt(s.substring(4,5));  
else  
    n2=Integer.parseInt(s.substring(3,5));  
System.out.println(n1*60+n2);  
}  
}
```

python implementation:

```
-----  
l=input().split(":")  
print(int(l[0])*60+int(l[1]))
```

LBP49

Next Prime

Given an integer,
create a function that returns the next prime.
If the number is prime, return the number itself.

input -----> a number

constraint -----> prime number

output -----> prime number

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int isprime(int n)
```

```
{
```

```
    int f=0,i;
```

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

```
    {
```

```
        if(n%i==0)
```

```
            f++;
```

```
    }
```

```
    return f==2;
```

```
}
```

```
int main() {
```

```
    int n;
```

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

```
    while(1)
```

```
{  
    if(isprime(n))  
    {  
        printf("%d",n);  
        break;  
    }  
    n++;  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static boolean isprime(int n)
```

```
    {
```

```
        int f=0,i;
```

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

```
        {
```

```
        if(n%i==0)

            f++;

    }

    return f==2;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n=obj.nextInt();

    while(true)

    {

        if(isprime(n))

        {

            System.out.println(n);

            break;

        }

        n++;

    }

}
```

python implementation:

```
def isprime(n):
```

```
f=0
```

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

```
    if n%i==0:
```

```
        f=f+1
```

```
return f==2
```

```
n=int(input())
```

```
while True:
```

```
    if isprime(n):
```

```
        print(n)
```

```
        break
```

```
n=n+1
```

LBP50

Sum of digits between two numbers

Create a function that sums the total number of digits between two numbers inclusive.

for example,

if the numbers are 19 and 22, then $(1+9)+(2+0)+(2+1)+(2+2)=19$.

input -----> a number from the user

constraints ----> no

output -----> sum of digits

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int sumofdigits(int n)
```

```
{
```

```
    int s=0;
```

```
    while(n!=0)
```

```
    {
```

```
        s=s+(n%10);
```

```
        n=n/10;
```

```
    }
```

```
    return s;
```

```
}
```

```
int main() {
```

```
    int n1,n2,s=0,i;
```

```
    scanf("%d %d",&n1,&n2);
```



```
for(i=n1;i<=n2;i++)  
{  
    s=s+sumofdigits(i);  
}  
printf("%d",s);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {  
    static int sumofdigits(int n)  
    {  
        int s=0;  
        while(n!=0)  
        {  
            s=s+(n%10);  
            n=n/10;  
        }  
        return s;  
    }  
}
```

```
}  
  
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    int n1=obj.nextInt(),n2=obj.nextInt(),i,s=0;  
  
    for(i=n1;i<=n2;i++)  
  
    {  
  
        s=s+sumofdigits(i);  
  
    }  
  
    System.out.println(s);  
  
}  
}
```

python implementation:

```
-----  
  
n1=int(input())  
n2=int(input())  
  
s=0  
  
for i in range(n1,n2+1):  
  
    s=s+sum([int(j) for j in str(i)])  
  
print(s)
```

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>


int sumOfDigits(int, int);


int main() {

    int n1,n2;

    scanf("%d %d",&n1,&n2);

    printf("%d",(sumOfDigits(n1, n2)));

    return 0;

}


int sumOfDigits(int n1, int n2){

    int sum=0,i;

    for(i=n1;i<=n2;i++){

        temp=i;

        while(temp!=0){

            sum=sum+temp%10;

            temp=temp/10;

        }

    }

}
```

```
return sum;  
}
```

LBP51

Defanging an IP address

Given a valid IP address, return a defanged version of that IP address.

A defanged IP address replaces every period '.' with "[.]".

input -----> A string

constraint -----> non-empty String

output -----> replacement String

. ==> [.]

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    char s[100],r[100]="\0";  
    int i,j=0;  
    scanf("%s",s);  
    for(i=0;s[i];i++)  
    {  
        if(s[i]=='.')  
        {  
            r[j++]='[';  
            r[j++]='.';  
            r[j++]=']';  
        }  
        else  
        {  
            r[j++] = s[i];  
        }  
    }  
    printf("%s",r);  
    return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        System.out.println(obj.nextLine().replace(".", "[.]"));  
    }  
}
```

python implementation:

```
-----  
  
s=input()  
print(s.replace('.', '[.]'))
```

LBP52

Jewels and Stones

You are given Strings jewels representing the types of stones that are jewels, and stones representing the stones you have. Each character in stones is a type

of stone you have you want to know how many of the stones you have are also jewels.

Letter are case sensitive. so "a" is considered as a different type of stone from "A".

input -----> A string

constraint -> no

output -----> how many of the stones

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char jw[100],s[100];
```

```
    int i,j,c=0;
```

```
    scanf("%s %s",jw,s);
```

```
    for(i=0;jw[i];i++)
```

```
    {
```

```
        for(j=0;s[j];j++)
```

```
{  
    if(jw[i]==s[j])  
        c++;  
}  
}  
printf("%d",c);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String jw=obj.nextLine();  
        String s=obj.nextLine();  
        int i,j,c=0;  
        for(i=0;i<jw.length();i++)  
        {
```



```
for(j=0;j<s.length();j++)  
{  
    if(jw.charAt(i)==s.charAt(j))  
        c++;  
}  
}  
System.out.println(c);  
}  
}
```

python implementation:

```
jw=input()  
s=input()  
c=0  
for i in jw:  
    c=c+s.count(i)  
print(c)
```

LBP53

Given a string s, and an integer array indices of the same length.

The string *s* will be shuffled such that the character at the *i*th position moves to indices[*i*] in the shuffled string, return shuffled string.

input -----> a string and an array

constraint ----> no

output -----> a string

s=aiohn

a=3 1 4 2 0

ts[*a*[*i*]]=*s*[*i*]

01234

nihao

ts[*a*[*i*]]=*s*[*i*]

i=0 ==> *ts*[*a*[0]]=*s*[0] ==> *ts*[3]=*a*

i=1 ==> *ts*[*a*[1]]=*s*[1] ==> *ts*[1]=*i* and so on

c implementation:

#include <stdio.h>

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100],ts[100]="\0";
```

```
    int i,a[100],n;
```

```
    scanf("%s",s);
```

```
    n=strlen(s);
```

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

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

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

```
        ts[a[i]]=s[i];
```

```
    printf("%s",ts);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        int i,n=s.length();  
        int a[]=new int[n];  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
        char b[]=new char[n];  
        for(i=0;i<n;i++)  
            b[a[i]]=s.charAt(i);  
        System.out.println(new String(b));  
    }  
}
```

python implementation:

```
s=input()  
a=[int(i) for i in input().split()]  
b=[0]*len(s)  
for i in range(0,len(s)):
```

```
b[a[i]]=s[i]  
print(''.join(b))
```

LBP54

Get word count

Problem Statement:

Create a function/method that takes a string and return the word count. The string will be a sentence.

Input: A string

Constraints: No

Output: Word count

c implementation:

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

```
int main() {
```

```
char s[100];

int i,c=1;

scanf("%[^\\n]s",s);

for(i=0;s[i];i++)

{

    if(s[i]==' ' && s[i+1]!=' ')

        c++;

}

printf("%d",c);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String[] s = obj.nextLine().split(" ");

        System.out.println(s.length);

    }

}
```

```
}  
  
}
```

python implementation:

```
s=input()  
L=s.split()  
print(len(L))
```

```
print(len(input().split()))
```

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj=new Scanner(System.in);  
        char[] ip=obj.next().toCharArray();  
        char[] defangedip=new char[30];
```

```
int i,j=0;
for(i=0;i<ip.length;i++){
    if(ip[i]=='.'){
        defangedip[j++]='[';
        defangedip[j++]='.';
        defangedip[j++]=']';
    }
    else
        defangedip[j++]=ip[i];
}
for(j=0;j<defangedip.length;j++)
    System.out.print(defangedip[j]);
}
}
```

LBP55

Check if String ending matches second String

Problem Statement: Create a function/method that takes two Strings and returns true if the first string ends with second string, otherwise return false.

Input: two strings

Constraints: No

Output: true or false

C implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int c=0,i,j;
```

```
    char s1[100],s2[100];
```

```
    scanf("%s %s",s1,s2);
```

```
    j=strlen(s1)-1;
```

```
    for(i=strlen(s2)-1;s2[i];i--)
```

```
{
```

```
    if(s2[i]==s1[j--])
```

```
        c++;
```

```
}  
  
if(c==strlen(s2))  
    printf("true");  
  
else  
    printf("false");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        String s1 = obj.nextLine();  
  
        String s2 = obj.nextLine();  
  
        System.out.println(s1.endsWith(s2));  
    }  
}
```

python implementation:

```
-----  
  
s1=input()  
s2=input()  
print("true" if s1.endswith(s2) else "false")
```

LBP56

Shuffle the Name

Problem Statement: Create a function/method that accepts a string (of person's first and last name) and returns a string with in first and last name swapped).

Input: two space separated strings

Constraints: No

Output: return last name followed by first name

abc xyz

xyz abc

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s1[100],s2[100];
```

```
    scanf("%s %s",s1,s2);
```

```
    printf("%s %s",s2,s1);
```

```
    return 0;
```

```
}
```

```
strcpy(t,s1);
```

```
strcpy(s1,s2);
```

```
strcpy(s2,t);
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
String s1 = obj.nextLine();  
  
String s2 = obj.nextLine();  
  
System.out.println(s2+" "+s1);  
  
}  
  
}
```

python implementation:

```
s1=input()  
s2=input()  
print(s2,s1)
```

LBP57

Reverse the order of a String

create a method/function that takes a string as its argument and returns the string in reversed order.

input -----> a string

constraint ----> no

output -----> reversed string

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100];

    scanf("%[^\\n]s",s);

    for(int i=strlen(s)-1;i>=0;i--)

        printf("%c",s[i]);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

StringBuffer sb = new StringBuffer(obj.nextLine());

System.out.println(sb.reverse());

}

}
```

```
String reverse(String obj)
{
    StringBuffer sb = new StringBuffer(obj);
    return sb.toString();
}
```

```
for(int i=s.length()-1;i>=0;i--)
{
    System.out.print(s.charAt(i));
}
```

python implementation:

```
-----
print(input()[::-1])
```

LBP58

Re-form the word

A word has been split into a left part and right part.

Re-form the word by adding both halves together changing the first to an uppercase letter.

input -----> two strings from the user

constraint ----> no

output -----> concatenated string with caps in first character

c implementation:

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

```
int main() {
    char s1[100],s2[100];
    scanf("%s %s",s1,s2);
    printf("%c",toupper(s1[0]));//C
    for(int i=1;s1[i];i++)
```



```
printf("%c",s1[i]);//omp  
  
printf("%s",s2);//lete  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s1 = obj.nextLine();  
  
        String s2 = obj.nextLine();  
  
        System.out.println(s1.substring(0,1).toUpperCase()+s1.substring(1)+s2);  
  
    }  
  
}
```

python implementation:

```
s1=input()
```

```
s2=input()  
print((s1+s2).title())
```

LBP59

Anagrams

Two strings a and b are called anagrams, if they contain all the same characters in the same frequencies.

input -----> two strings a and b

constraint ---> no

output -----> true or false

c implementation:

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

```
int main() {  
    char s1[100],s2[100],ch;
```

```
int i,j;

scanf("%s %s",s1,s2);

for(i=0;s1[i];i++)
{
    for(j=i+1;s1[j];j++)
    {
        if(s1[i]>s1[j]){
            ch = s1[i];
            s1[i]=s1[j];
            s1[j]=ch;
        }
    }
}

for(i=0;s2[i];i++)
{
    for(j=i+1;s2[j];j++)
    {
        if(s2[i]>s2[j]){
            ch = s2[i];
            s2[i]=s2[j];
            s2[j]=ch;
        }
    }
}
```

```
}  
  
if(strcmp(s1,s2)==0)  
    printf("true");  
  
else  
    printf("false");  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        char ch1[] = obj.nextLine().toCharArray();  
        char ch2[] = obj.nextLine().toCharArray();  
        Arrays.sort(ch1);  
        Arrays.sort(ch2);  
        System.out.println(Arrays.equals(ch1,ch2));  
    }  
}
```

```
}
```

python implementation:

```
s1=input()
```

```
s2=input()
```

```
print("true" if sorted(s1)==sorted(s2) else "false")
```

LBP60

Max Occurring Character

Given a string, implement a program to find max occurring character in the given string

input -----> A string from the user.

constraints--> No

output -----> max occurring character

welcome ----> e

java -----> a

```
a[256]={0}
```

```
for(i=0;s[i];i++)
```

```
    a[(int)s[i]]++;
```

```
a[w]=1
```

```
a[e]=1=>2
```

```
a[l]=1
```

```
a[c]=1
```

```
a[o]=1
```

```
a[m]=1
```

```
max=0;
```

```
for(i=0;i<256;i++)
```

```
{
```

```
    max element
```

```
    m=i;
```

```
}
```

```
max=2
```

```
print m
```

c implementation:

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

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

```
#include <math.h>

#include <stdlib.h>

int main() {

    char s[100];

    int i,a[256]={0},max;

    scanf("%s",s);

    for(i=0;s[i];i++)

        a[(int)s[i]]++;

    max=0;

    for(i=0;i<256;i++)

    {

        if(a[i]>a[max])

            max=i;

    }

    printf("%c",max);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int a[]=new int[256];  
  
        int i,max=0;  
  
        for(i=0;i<s.length();i++)  
  
            a[(int)s.charAt(i)]++;  
  
        max=0;  
  
        for(i=0;i<256;i++)  
  
        {  
  
            if(a[i]>a[max])  
  
                max=i;  
  
        }  
  
        System.out.println((char)max);  
  
    }  
}
```

python implementation:

```
from collections import Counter
```



```
s=input()
r=Counter(s)
#print(r)
print(max(r,key=r.get))
```

LBP61

Determine the color of a chess board square

You are given coordinates, a string that represents the coordinates of a square of the chess board. bellow is the chess board for your reference.

Return True if the saquare is in white, and false if the square is in Black.

The coordinates will always represent a valid chess board square. The coordinates will always have the letter first, and the number second.

input -----> a string

constraint -----> length of the string is 2, $a \leq c[0] \leq h$ and $1 \leq c[1] \leq 8$

output -----> true or false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[10];
```

```
    int x,y;
```

```
    scanf("%s",s);
```

```
    x=s[0]-96;
```

```
    y=s[1];
```

```
    if(x%2!=y%2)
```

```
        printf("true");
```

```
    else
```

```
        printf("false");
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    String s = obj.nextLine();  
    int x = s.charAt(0)-96;  
    int y = s.charAt(1);  
    System.out.println(x%2!=y%2);  
}  
}
```

python implementation:

```
-----  
s=input()  
x=ord(s[0])-96  
y=ord(s[1])  
print(str(x%2!=y%2).lower())
```

LBP62

Find the Bomb

Write a function that finds the word "bomb" in the given string (not case sensitive)

return "DUCK!" if found, else return "Relax there's no bomb."

input -----> a string

constraint ----> no

output -----> "DUCK!" or "Relax, There's no bomb."

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s1[100],s2[]="bomb";
```

```
    scanf("%[^\\n]s",s1);
```

```
    int i,j,k,len,found=0;
```

```
    for(i=0;s1[i];i++)
```

```
    {
```

```
        if(s1[i]>='A' && s1[i]<='Z')
```

```
            s1[i]=s1[i]+32;
```

```
    }
```

```
len=strlen(s2);
for(i=0;s1[i];i++)
{
    k=i;
    for(j=0;j<len;j++)
    {
        if(s1[k]!=s2[j])
            break;
        k++;
    }
    if(j==len)
        found=1;
}
printf((found==1)?"DUCK!":"Relax, there's no bomb.");
return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine().toLowerCase();  
        System.out.println((s.contains("bomb"))?"DUCK!":"Relax, there's no  
bomb.");  
    }  
}
```

python implementation:

```
-----  
s=input().lower()  
if "bomb" in s:  
    print("DUCK!")  
else:  
    print("Relax, there's no bomb.")
```

LBP63

How many vowels

Create a function that takes a string and returns the number of vowels contained within it.

input -----> a string

constraint -----> no

output -----> number of vowels

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,c=0;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
```

```
            c++;
```

```
    }
```

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

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        int i=0,c=0;
```

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

```
        {
```

```
            if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' || s.charAt(i)=='o' || s.charAt(i)=='u')
```

```
                c++;
```

```
        }
```

```
        System.out.println(c);
```

```
    }
```

```
}
```


python implementation:

```
s=input()
```

```
c=0
```

```
for i in s:
```

```
    if i in "aeiou":
```

```
        c=c+1
```

```
print(c)
```

LBP64

X's and O's, Nobody knows

Create a function that takes a string,

check if it has the same number of x's and o's and returns either true or false.

Rules:

1. return boolean value true or false.
2. returns true if the amount x's and o's are the same.
3. returns false if they are not the same amount.
4. the string can contains any character.
5. when 'x' and 'o' are not in the string, return true.

input -----> a string

constraints----> no

output -----> true or false

xc=0

oc=0

logic for xc and oc

if(xc==oc) then true else false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    scanf("%s",s);
```

```
int i,xc=0,oc=0;

for(i=0;s[i];i++)
{
    if(s[i]=='x')
        xc++;
    if(s[i]=='o')
        oc++;
}

printf((xc==oc)?"true":"false");

return 0;
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

        int i,xc=0,oc=0;
```

```
for(i=0;i<s.length();i++)  
{  
    if(s.charAt(i)=='x')  xc++;  
    if(s.charAt(i)=='o')  oc++;  
}  
System.out.println(xc==oc);  
}  
}
```

python implementation:

```
s=input()  
print("true" if s.count('x')==s.count('o') else "false")
```

LBP65

Stuttering Function

write a function that shutters a word as if someone is struggling to read it.

The first two letters are repeated twice with an ellipsis ... , and then the word is pronounced with a question mark?

input -----> a string

contraint -----> no

output -----> xx...xx...~~~~~?

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    scanf("%s",s);
```

```
    printf("%c%c...",s[0],s[1]);
```

```
    printf("%c%c...",s[0],s[1]);
```

```
    printf("%s?",s);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        System.out.print(s.substring(0,2)+"..." +s.substring(0,2)+"..." +s+"?");  
  
    }  
}
```

python implementation:

```
s=input()  
print(f"{s[0]}{s[1]}...{s[0]}{s[1]}...{s}?")
```

LBP66

Repeating Letters

Create a method that takes a string and returns a string in which each character is repeated once.

input -----> String from the user

constraint -----> No

output -----> String

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        printf("%c%c",s[i],s[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

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

        {

            System.out.print(s.charAt(i)+" "+s.charAt(i));

        }

    }

}
```

python implementation:

```
s=input()

for i in s:

    print(i*2,end="")
```

LBP67

Double Letters

Create a function that takes a word and returns true if the word has two consecutive identical letters.

input -----> A string

constraint-----> No

output -----> true or false

c implementation:

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

```
int main() {
    char s[100];
    int i,found=0;
    scanf("%s",s);
    for(i=0;s[i];i++)
    {
        if(s[i]==s[i+1]){
```

```
        found=1;

        break;

    }

}

printf((found==1)?"true":"false");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

        int i;

        boolean found=false;

        for(i=0;i<s.length()-1;i++)

        {

            if(s.charAt(i)==s.charAt(i+1))
```

```
{  
    found=true;  
    break;  
}  
}  
System.out.println(found);  
}
```

python implementation:

```
-----  
s=input()  
found=False  
for i in range(len(s)-1):  
    if s[i]==s[i+1]:  
        found=True  
        break  
print(str(found).lower())
```

True ----> true

False ---> false

str(True).lower()

LBP68

Andy, Ben and Charlotte are playing a board game.

The three of them decided to come up with a new scoring system.

A player's first initial ("A", "B" & "C") denotes that players scoring a single point.

Given a string of capital letters. returns an array of the player's scores.

input -----> A String

constraint -----> No

output -----> score

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,a=0,b=0,c=0;
```

```
scanf("%s",s);  
for(i=0;s[i];i++)  
{  
    if(s[i]=='A') a++;  
    if(s[i]=='B') b++;  
    if(s[i]=='C') c++;  
}  
printf("%d %d %d",a,b,c);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        int a=0,b=0,c=0,i;  
        for(i=0;i<s.length();i++)
```

```
{  
    if(s.charAt(i)=='A') a++;  
    if(s.charAt(i)=='B') b++;  
    if(s.charAt(i)=='C') c++;  
}  
System.out.println(a+" "+b+" "+c);  
}  
}
```

python implementation:

```
-----  
s=input()  
print(s.count('A'),s.count('B'),s.count('C'))
```

LBP69

Remove Every vowel from a String

Create a function that takes a string and returns a new string with all vowels removed.

input -----> a string

constraints -----> No

output -----> a string

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
```

```
            continue;
```

```
        else
```

```
            printf("%c",s[i]);
```

```
    }
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        System.out.println(obj.nextLine().replaceAll("[aeiou]", ""));
```

```
    }
```

```
}
```

python implementation:

```
import re
```

```
print(re.sub("[aeiou]", "", input()))
```

LBP70

Space between each character

Create a function that takes a string and returns a string with spaces in between all of the characters.

input -----> a string

constraints-----> No

output -----> string

c implementation:

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

```
int main() {
    char s[100];
    int i;
    scanf("%s",s);
    for(i=0;s[i];i++)
        printf("%c ",s[i]);
    return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

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

```
        {
```

```
            System.out.print(s.charAt(i)+" ");
```

```
        }
```

```
    }
```

```
}
```

python implementation:

```
s=input()
```

```
for i in s:
```

```
    print(i,end=' ')
```

```
print(' '.join([i for i in input()])))
```

LBP71

VOWEL REPLACER

Create a function that replaces all the vowels in a string with a specified character,

input -----> A string from the user and a character

cons -----> no

output -----> A string

```
for(int i=0;s[i];i++)  
{  
    if(s[i] is vowel)  
        print char  
    else  
        print s[i]  
}
```

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100],ch;

    scanf("%s",s);

    scanf("\n%c",&ch);

    for(int i=0;s[i];i++)

    {

        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')

            printf("%c",ch);

        else

            printf("%c",s[i]);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

        String ch = obj.nextLine();

        System.out.println(s.replaceAll("[aeiou]",ch));

    }

}
```

python implementation:

```
import re

s1=input()

s2=input()

print(re.sub("[aeiou]",s2,s1))
```

LBP72

Say "Hello" Say "Bye"

Write a function that takes a string name and number num (either 1 or 0) and return "Hello"+name if number is 1, otherwise "Bye"+name.

input -----> a string from the user

constraint -> no

output -----> a string

c implementation:

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

```
int main() {
    char s[100];
    int n;
    scanf("%s",s);
    scanf("%d",&n);
    if(n==1)
        printf("Hello %s",s);
```

```
else  
  
    printf("Bye %s",s);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int n = obj.nextInt();  
  
        if(n==1)  
  
            System.out.println("Hello "+s);  
  
        else  
  
            System.out.println("Bye "+s);  
  
    }  
  
}
```

python implementation:

```
s=input()
n=int(input())
if n==1:
    print("Hello",s)
else:
    print("Bye",s)
```

LBP73

VALID ZIP CODE

zipcodes consists of 5 consecutive digits.

Given a string,

write a function to determine whether the input is a valid zip code.

a valid zipcode is as follows

1. must contain only numbers.
2. it should not contain any spaces.
3. length should be only 5.

input -----> A string

constraint -> no

output -----> true or false

c=0

for(i=0;s[i];i++)

{

if(s[i]>='0' and s[i]<='9')

c++;

}

print c==5 print true else false

c implementation:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

char s[100];

int i,c=0;

scanf("%[^\n]s",s);

```
for(i=0;s[i];i++)  
{  
    if(s[i]>='0' && s[i]<='9')  
        c++;  
}  
printf((c==5)?"true":"false");  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        try  
        {  
            int n=Integer.parseInt(s);
```

System.out.println(n>9999 && n<100000);//Harshit Nikam

```
}  
  
catch(Exception e)  
  
{  
  
    System.out.println(false);  
  
}  
  
}  
  
}
```

python implementation:

```
s=input()  
  
c=0  
  
for i in s:  
  
    if i.isdigit():  
  
        c=c+1  
  
print("true" if c==5 else "false")
```

LBP74

Returns the middle character of a string

create a function that takes a string and returns, the middle character(s).

if the word's length is odd return the middle character.

if the word's length is even, return the middle two characters.

input ----> a string from the user

constraint-> no

output ----> middle character(s)

if length is even print $s[n/2-1], s[n/2]$ else print $s[n/2]$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int n;
```

```
    scanf("%s",s);
```

```
    n=strlen(s);
```

```
    if(n%2==0)
```

```
printf("%c%c",s[n/2-1],s[n/2]);  
  
else  
  
printf("%c",s[n/2]);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int n = s.length();  
  
        if(n%2==0)  
  
            System.out.print(s.charAt(n/2-1)+" "+s.charAt(n/2));  
  
        else  
  
            System.out.println(s.charAt(n/2));  
  
    }  
  
}
```

python implementation:

```
s=input()
```

```
n=len(s)
```

```
if n%2==0:
```

```
    print(s[n//2-1],s[n//2],sep='')
```

```
else:
```

```
    print(s[n//2])
```

LBP75

Index of first vowel

create a function that returns the index of first vowel in a string

input -----> a string

con -----> no

output -----> an int value

logic:

```
for(i=0;s[i];i++)
```

```
{  
    if(s[i] aeiou)  
    {  
        print i  
        break  
    }  
}
```

c implementation:

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

```
int main() {  
    char s[100];  
    int i;  
    scanf("%s",s);  
    for(i=0;s[i];i++)  
    {  
        if(s[i]=='a' | | s[i]=='e' | | s[i]=='i' | | s[i]=='o' | | s[i]=='u')
```

```
{  
    printf("%d",i);  
    break;  
}  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

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

```
        {
```

```
            if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' || s.charAt(i)=='o' || s.charAt(i)=='u')
```

```
            {
```



```
        System.out.println(i);  
        break;  
    }  
}  
}
```

python implementation:

```
s=input()  
for i in range(0,len(s)):  
    if s[i] in "aeiou":  
        print(i)  
        break
```

LBP76

Longest Word

Write a function that finds the longest word in a sentence.

If two or more words are found, return the first longest word.

Characters such as apostrophe, comma, period (and the like) count as part of the word

(e.g. O'Connor is 8 characters long).

input -----> a string from the user

con -----> no

output-----> longest word

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100],*p,res[100]="\0";
```

```
    int m;
```

```
    scanf("%[^\n]s",s);
```

```
    p=strtok(s," ");
```

```
    m=0;
```

```
    while(p!=NULL){
```

```
        if(strlen(p)>m){
```

```
m=strlen(p);  
strcpy(res,p);  
}  
p=strtok(NULL," ");  
}  
printf("%s",res);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        StringTokenizer st = new StringTokenizer(s);  
        int m=0;  
        String res="";  
        while(st.hasMoreTokens())
```

```
{  
    String token = st.nextToken();  
    if(token.length()>m)  
    {  
        m=token.length();  
        res=token;  
    }  
}  
System.out.println(res);  
}
```

python implementation:

```
L=input().split()  
m=0  
s=""  
for i in L:  
    if len(i)>m:  
        m=len(i)  
        s=i  
print(s)
```

LBP77

Print all permutations of a string

Given a string str, the task is to print all the permutations of str.

A permutation is an arrangement of all or part of a set of objects, with regard to the order of the arrangement.

For instance, the words "bat" and "tab" represents two distinct permutation (or arrangements)

of a similar three letter word.

input ----> string from the user

con -----> no

output ---> all permutations of the string

A [1!=1] -----> A

AB [2!=2] -----> AB, BA

ABC [3!=6] -----> ABC, ACB, BAC, BCA, CAB, CBA

ABCD [4!=24] ----->

ABCD ABDC ACBD ACDB ADCB ADBC BACD BADC BCAD BCDA BDCA BDAC CBAD
CBDA CABD CADB CDAB CDBA DBCA DBAC DCBA DCAB DACB DABC

ABCDE [5!=120] ---->

ABCDE ABCED ABDCE ABDEC ABEDC ABECD ACBDE ACBED ACDBE ACDEB
ACEDB ACEBD ADCBE ADCEB ADBCE ADBEC ADEBC ADECB AECDB AECBD
AEDCB AEDBC AEBDC AEBDC BACDE BACED BADCE BADEC BAEDC BAEDC
BCADE BCAED BCDAE BCDEA BCEDA BCEAD BDCAE BDCEA BDACE BDAEC
BDEAC BDECA BECDA BECAD BEDCA BEDAC BEADC BEACD CBADE CBAED
CBDAE CBDEA CBEDA CBEAD CABDE CABED CADBE CADEB CAEDB CAEBD
CDABE CDAEB CDBAE CDBEA CDEBA CDEAB CEADB CEABD CEDAB CEDBA
CEBDA CEBAD DBCAE DBCEA DBACE DBAEC DBEAC DBECA DCBAE DCBEA
DCABE DCAEB DCEAB DCEBA DACBE DACEB DABCE DABEC DAEBE DAECB
DECAB DECBA DEACB DEABC DEBAC DEBCA EBCDA EBCAD EBDCA EBDAC
EBADC EBACD ECBDA ECBAD ECDBA ECDAB ECADB ECABD EDCBA EDCAB
EDBCA EDBAC EDABC EDACB EACDB EACBD EADCB EADBC EABDC EABCD

6==> {1,5}, {2,4}, {3,3}, {3,3}, {4,2} and {5,1}

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
void swap(char *x,char *y){
```

```
    char t;
```

```
    t=*x;
```

```
    *x=*y;
```

```
*y=t;

}

void permute(char *p,int left,int right){

    int i;

    if(left==right)

        printf("%s ",p);

    else{

        for(i=left;i<=right;i++){

            swap((p+left),(p+i));

            permute(p,left+1,right);

            swap((p+left),(p+i));

        }

    }

}
```

```
int main() {

    char s[100];

    scanf("%s",s);

    permute(s,0,strlen(s)-1);

    return 0;

}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void permute(String s,int left,int right){  
  
        if(left==right)  
  
            System.out.print(s+" ");  
  
        else{  
  
            for(int i=left;i<=right;i++){  
  
                s = swap(s,left,i);  
  
                permute(s,left+1,right);  
  
                s = swap(s,left,i);  
  
            }  
  
        }  
    }  
  
    static String swap(String a,int i,int j){  
  
        char temp;  
  
        char[] charArray=a.toCharArray();  
  
        temp = charArray[i];  
  
        charArray[i]=charArray[j];  
  
        charArray[j]=temp;  
  
        return String.valueOf(charArray);  
    }  
}
```



```
}  
  
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    String s = obj.nextLine();  
  
    permute(s,0,s.length()-1);  
  
}  
}
```

python implementation:

```
from itertools import permutations  
  
l=list(permutations(input()))  
  
for i in l:  
  
    print("".join(i),end=' ')
```

LBP78

Removing Duplicate Characters from a string

Given a string S, the task is to remove all the duplicates in the given string.

input -----> a string from the user

con -----> remove all duplicates

output -----> a string without duplicates

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,j,k;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++){
```

```
        for(j=i+1;s[j];j++){
```

```
            if(s[i]==s[j]){
```

```
                for(k=j;s[k];k++)
```

```
                    s[k]=s[k+1];
```

```
            }
```

```
        }
```

```
    }
```

```
    printf("%s",s);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        String rs="";
```

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

```
        {
```

```
            char ch = s.charAt(i);
```

```
            if(rs.indexOf(ch)<0)
```

```
                rs=rs+ch;
```

```
        }
```

```
        System.out.println(rs);
```

```
    }
```

```
}
```

python implementation:

```
s=input()
```

```
l=[]
```

```
for i in s:
```

```
    if i not in l:
```

```
        l.append(i)
```

```
print(''.join(l))
```

LBP79

Swap corner words and reverse middle characters

Write a Java program to take an input string and exchange the first and last word and reverse the middle word.

input -----> a string

con -----> no

output -----> a string

Hello welcome to java ==> java ot emoclew Hello

java is very easy ==> easy yrev si java

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100],fw[100]="\0",lw[100]="\0";
```

```
    int i,j,k;
```

```
    scanf("%[^\\n]s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        k=0;
```

```
        while(s[i]!=' '){
```

```
            fw[k++]=s[i++];
```

```
        }
```

```
        break;
```

```
    }
```

```
    for(j=strlen(s)-1;s[j];j--)
```

```
    {
```

```
        k=0;
```

```
        while(s[j]!=' '){
```

```
lw[k++]=s[j--];  
  
}  
  
break;  
  
}  
  
for(k=strlen(lw)-1;k>=0;k--)  
    printf("%c",lw[k]);  
  
printf(" ");  
  
for(k=j-1;k>=i;k--)  
    printf("%c",s[k]);  
  
printf("%s",fw);  
  
return 0;  
  
}
```

abc xyz mno pqr

pqr onm zyx abc

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s[] = obj.nextLine().split(" ");  
        System.out.print(s[s.length-1]+" ");  
        for(int i=s.length-2;i>=1;i--)  
        {  
            System.out.print(new StringBuffer(s[i]).reverse()+" ");  
        }  
        System.out.print(s[0]);  
    }  
}
```

python implementation:

```
-----  
l=input().split()  
print(l[-1],end=' '  
for i in range(len(l)-2,0,-1):  
    print(l[i][::-1],end=' '  
print(l[0])
```

LBP80

Valid Hex Code

Create a function that determines whether a string is a valid hex code.

A hex code must begin with a pound key # and is exactly 6 characters in length.

Each character must be a digit from 0-9 or an alphabetic character from A-F.

All alphabetic characters may be uppercase or lowercase.

input ----> a string from the user

con -----> no

output ----> true or false

#XXXXXX ----> 7

0-9 and A-F

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```



```
char s[100];

scanf("%s",s);

int i,c=0;

for(i=0;s[i];i++)

{

    if(s[i]>='a' && s[i]<='z')

        s[i]=s[i]-32;

}

if(s[0]=='#' && strlen(s)==7)

{

    for(i=1;s[i];i++)

    {

        if((s[i]>='A' && s[i]<='F') || (s[i]>='0' && s[i]<='9'))

            c++;

    }

}

printf((c==6)?"true":"false");

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        System.out.println(obj.nextLine().matches("#[A-Fa-f0-9]{6}"));

    }

}
```

python implementation:

```
import re

print("true" if re.fullmatch("#[A-F0-9a-f]{6}",input())!=None else "false")
```

LBP81

Even Length Words

Write a program to print even length words in a string?

input -----> a string from the user

con -----> no

output ----> list of strings with even length

string tokenization

get token by token and then check whether its length is divisible by 2 or not.

c implementation:

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

```
int main() {
    char s[100], *p;
    scanf("%[^\\n]s", s);
    p = strtok(s, " ");
    while(p != NULL) {
        if(strlen(p) % 2 == 0)
            printf("%s ", p);
        p = strtok(NULL, " ");
    }
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String[] s = obj.nextLine().split(" ");
```

```
        for(String str:s){
```

```
            if(str.length()%2==0)
```

```
                System.out.print(str+" ");
```

```
        }
```

```
    }
```

```
}
```

python implementation:

```
s=input()
```

```
l=s.split()
```

for i in l:

if len(i)%2==0:

print(i,end=' ')

```
static String evenLength(String s){
```

```
    String ss[]=s.split(" "),sss="";
```

```
    for(String t:ss){
```

```
        if(t.length()%2==0){
```

```
            sss+=t+" ";
```

```
        }
```

```
    }
```

```
    return sss;
```

```
}
```

LBP82

Change Every Letter to the Next Letter

Write a function that changes every letter to the next letter:

"a" becomes "b"

"b" becomes "c"

"d" becomes "e"

and so on ...

note: there is no z's in test cases, be happy.

input -----> a string from the user

cons -----> no

output -----> modified string

logic:

s[i]+1

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    scanf("%s",s);
```

```
for(int i=0;s[i];i++){  
    if(s[i]!='z')  
        s[i]=s[i]+1;  
}  
printf("%s",s);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        for(int i=0;i<s.length();i++){  
            System.out.print((char)(s.charAt(i)+1));  
        }  
    }  
}
```

python implementation:

```
-----  
  
s=input()  
  
for i in s:  
    print(chr(ord(i)+1),end="")
```

LBP83

First N Vowels

Write a function that returns the first n vowels of a string.

input -----> a string from the user and an integer value

cons -----> Return "invalid" if the n exceeds the number of vowels in a string.

output -----> return first n vowels in the string

c implementation:

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



```
#include <math.h>

#include <stdlib.h>

int main() {

    char s[100],ns[100]="\0";

    int i,j=0,n;

    scanf("%[^\\n]s",s);

    scanf("%d",&n);

    for(i=0;s[i];i++){

        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')

            ns[j++]=s[i];

    }

    if(n>strlen(ns))

        printf("invalid");

    else{

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

            printf("%c",ns[i]);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

        int n=obj.nextInt();

        StringBuffer sb = new StringBuffer();

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

            char ch=s.charAt(i);

            if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')

                sb.append(ch);

        }

        if(n>sb.length())

            System.out.println("invalid");

        else{

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

                System.out.print(sb.charAt(i));

            }

        }

    }

}
```

}

python implementation:

```
import re
s=input()
n=int(input())
ns=re.sub("[^aeiouAEIOU]", "", s)
print("invalid" if n>len(ns) else ns[0:n])
```

LBP84

Is the String in Order?

Create a function that takes a string and returns true or false,
depending on whether the characters are in order or not.

input -----> a string from the user

cons -----> for non-empty string print invalid

output -----> true or false

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
  
    char s1[100],s2[100]="\0";  
  
    scanf("%s",s1);  
  
    strcpy(s2,s1);  
  
    for(int i=0;s1[i];i++){  
        for(int j=i+1;s1[j];j++){  
            if(s1[i]>s1[j]){  
                char ch;  
                ch=s1[i];  
                s1[i]=s1[j];  
                s1[j]=ch;  
            }  
        }  
    }  
  
    printf((strcmp(s1,s2)==0)?"true":"false");  
  
    return 0;  
  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s1 = obj.nextLine();
```

```
        char ch[]=s1.toCharArray();
```

```
        Arrays.sort(ch);
```

```
        String s2 = new String(ch);
```

```
        System.out.println(s1.equals(s2));
```

```
    }
```

```
}
```

python implementation:

```
s1=input()
```

```
l=list(s1)
```

```
l.sort()  
s2="".join(l)  
print("true" if s1==s2 else "false")
```

LBP85

Integer to English Words

Convert a non-negative integer num to its English words representation.

input -----> a number from the user
con -----> n>0
output -----> number in English words

123 ==> one hundred twenty three

9999 ==> Nine Thousand Nine Hundred Ninty Nine

c implementation:

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

```
#include <string.h>

#include <math.h>

#include <stdlib.h>


int main() {

    char
    *belowten[]={"Zero","One","Two","Three","Four","Five","Six","Seven","Eight",
    "Nine"};

    char
    *tensplace[]={"","Ten","Twenty","Thirty","Fourty","Fifty","Sixty","Seventy","Ei
    ghty","Ninety"};

    char
    *belowtwenty[]={"","Ten","Eleven","Twelve","Thirteen","Fourteen","Fifteen","
    Sixteen","Seventeen","Eighteen","Nineteen"};

    char s[10];

    scanf("%s",s);

    int num,len;

    len=strlen(s);

    if(len==1){

        num=s[0]-48;

        printf("%s ",belowten[num]);

    }

    else if(len==2 && s[1]==48){ //10,20,30,40,50,60,70,80,90

        num=s[0]-48;

        printf("%s ",tensplace[num]);

    }

}
```

```
}  
  
else if(len==2 && s[0]==49){ //10,11,12,13,14,15,16,17,18,19  
  
    num=s[1]+1-48;  
  
    printf("%s ",belowtwenty[num]);  
  
}  
  
else if(len==2){ //21-99  
  
    num = s[0]-48;  
  
    printf("%s ",tensplace[num]);  
  
    num=s[1]-48;  
  
    printf("%s ",belowten[num]);  
  
}  
  
else if(len==3){  
  
    num=s[0]-48;  
  
    printf("%s ",belowten[num]);  
  
    printf("Hundred ");  
  
    if(s[1]==48){//101,102,203  
  
        num=s[2]-48;  
  
        printf("%s ",belowten[num]);  
  
    }  
  
    else if(s[1]==49){//112,219,..  
  
        num=s[2]+1-48;  
  
        printf("%s ",belowtwenty[num]);  
  
    }  
  
}
```



```
else if(s[2]==48){//x10, x50, x90

    num=s[1]-48;

    printf("%s ",tensplace[num]);

}

else{ //856

    num=s[1]-48;

    printf("%s ",tensplace[num]);

    num=s[2]-48;

    printf("%s ",belowten[num]);

}

}

else if(len==4){

    num=s[0]-48;

    printf("%s ",belowten[num]);

    printf("Thousand ");

    if(s[1]-48!=0){

        num=s[1]-48;

        printf("%s ",belowten[num]);

        printf("Hundred ");

    }

    if(s[2]==48){

        num=s[3]-48;

        if(num!=0)
```

```
        printf("%s ",belowten[num]);  
    }  
    else  
    {  
        if(s[3]==48){  
            num=s[2]-48;  
            printf("%s ",tensplace[num]);  
        }  
        else if(s[2]==49){  
            num=s[3]-48+1;  
            printf("%s ",belowtwenty[num]);  
        }  
        else{  
            num=s[2]-48;  
            printf("%s ",tensplace[num]);  
            num=s[3]-48;  
            printf("%s ",belowten[num]);  
        }  
    }  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static String[]
```

```
    belowten={"","One","Two","Three","Four","Five","Six","Seven","Eight","Nine"}
    ;
```

```
    static String[]
```

```
    belowtwenty={"Ten","Eleven","Twelve","Thirteen","Fourteen","Fifteen","Sixteen",
    "Seventeen","Eighteen","Nineteen"};
```

```
    static String[]
```

```
    belowhundred={"","Ten","Twenty","Thirty","Fourty","Fifty","Sixty","Seventy","
    Eighty","Ninety"};
```

```
    static String helpme(int n){
```

```
        String result="";
```

```
        if(n<10){
```

```
            result=belowten[n];
```

```
        }
```

```
        else if(n<20){
```

```
            result=belowtwenty[n];
```

```
}  
  
else if(n<100){  
    result=belowhundred[n/10]+" "+helpme(n%10);  
}  
  
else if(n<1000){  
    result=helpme(n/100)+" Hundred "+helpme(n%100);  
}  
  
else if(n<10000)  
    result=helpme(n/1000)+" Thousand "+helpme(n%1000);  
  
return result;  
}
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
  
    int n=obj.nextInt();  
  
    if(n==0)  
        System.out.println("Zero");  
  
    else  
        System.out.println(helpme(n));  
}  
}
```

python implementation:

```
-----  
d1=["","One","Two","Three","Four","Five","Six","Seven","Eight","Nine","",  
"Eleven","Twelve","Thirteen","Fourteen","Fifteen","Sixteen","Seventeen","Eighteen",  
,"Nineteen"]
```

```
d2=["","","Twenty","Thirty","Fourty","Fifty","Sixty","Seventy","Eighty","Ninety",  
]
```

```
def check(n,string):
```

```
    if n==0:
```

```
        return "
```

```
    elif n>=19:
```

```
        s=d2[n//10]+' '+d1[n%10]+string
```

```
        return s
```

```
    else:
```

```
        return d1[n]+' '+string
```

```
n=int(input())
```

```
if n==0:
```

```
    print('zero')
```

```
else:
```

```
    result=check((n//1000)%100,'Thousand ')
```

```
    result+=check((n//100)%10,'Hundred ')
```

```
    result+=check(n%100,"")
```

```
    print(result)
```

sir will u start dsa course in future plz sir confirm it

- 1) Full Stack Python Developer
- 2) DSA
- 3) Advanced LBP

LBP86

C*ns*r*d Str*ngs

Someone has attempted to censor my strings by replacing every vowel with a *, l*k* th*s.

Luckily, I've been able to find the vowels that were removed.

Given a censored string and a string of the censored vowels, return the original uncensored string.

input -----> original & replacement strings

con -----> no

output -----> updated string after replacement

Ex:

w*lc*m*,eoe ----> welcome

```
for(i=0;s[i];i++){  
    if(s[i]=='*')  
        print ss[j++]  
    else  
        print s[i]
```

c implementation:

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

```
int main() {  
    char s[100],ss[100];  
    scanf("%[^\\n]s",s);  
    scanf("%s",ss);  
    int i,j=0;  
    for(i=0;s[i];i++){  
        if(s[i]=='*')  
            printf("%c",ss[j++]);
```

```
else  
  
    printf("%c",s[i]);  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        String ss = obj.nextLine();  
  
        int i,j=0;  
  
        for(i=0;i<s.length();i++){  
  
            if(s.charAt(i)=='*')  
  
                System.out.print(ss.charAt(j++));  
  
            else  
  
                System.out.print(s.charAt(i));  
  
        }  
    }  
}
```



```
}  
  
}  
  
}
```

python implementation:

```
-----  
  
s=input()  
ss=input()  
  
j=0  
for i in s:  
    if i=='*':  
        print(ss[j],end='')  
        j=j+1  
    else:  
        print(i,end='')
```

LBP87

parentheses balance

Given a string S of '(' and ')' parentheses,
we add the minimum number of parentheses ('(' or ')', and in any positions)
so that the resulting parentheses string is valid.

Formally, a parentheses string is valid if and only if:

It is the empty string, or It can be written as AB (A concatenated with B), where A and B are

valid strings, or It can be written as (A), where A is a valid string.

Given a parentheses string,

return the minimum number of parentheses we must add to make the resulting string valid.

input -----> a string from the user

con -----> no

output -----> number of parentheses to be added

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
int i,c=0;

scanf("%s",s);

for(i=0;s[i];i++){

    if(s[i]=='(') c++;

    if(s[i]==')') c--;

}

printf("%d",c);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();

        int i,c=0;

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

            if(s.charAt(i)=='(') c++;

        }

    }

}
```

```
        if(s.charAt(i)=='') c--;  
    }  
    System.out.println(c);  
}  
}
```

python implementation:

```
s=input()  
c=0  
for i in s:  
    if i=='(':  
        c=c+1  
    if i==')':  
        c=c-1  
print(c)
```

LBP88

American keyboard

Given a string, return the true if that can be typed using letters of alphabet on only

one row's of American keyboard like the image below.

In the American keyboard:

the first row consists of the characters "qwertyuiop",

the second row consists of the characters "asdfghjkl", and

the third row consists of the characters "zxcvbnm".

dad ---> true

mom ---> false

Note:

1. You may use one character in the keyboard more than once.
2. You may assume the input string will only contain letters of alphabet.

input -----> a string from the user

cons -----> no

output -----> true or false

logic:

```
for(i=0;s[i];i++){
```

```
    s[i]==r1[j]    c1++
```

```
    s[i]==r2[j]    c2++
```

```
    s[i]==r3[j]    c3++
```

```
}
```

```
dad->3
```

```
c2=0->1->2->3
```

```
c1=0
```

```
c3=0
```

```
mom->3
```

```
c3=0->1->2
```

```
c1=0->1
```

```
c2=0
```

You made every problem's solution very very easy sir

7 yrs of experience as Java developer today I got palindrome question

just for practice habit we joined u r class .

it's making as regular habit do a programming

thanks for gd questions

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100],r1[100]="qwertyuiop",r2[100]="asdfghjkl",r3[100]="zxcvbnm";

    int j,i,c1=0,c2=0,c3=0;

    scanf("%s",s);

    for(i=0;s[i];i++){

        for(j=0;r1[j];j++){

            if(s[i]==r1[j]) c1++;

        }

        for(j=0;r2[j];j++){

            if(s[i]==r2[j]) c2++;

        }

        for(j=0;r3[j];j++){

            if(s[i]==r3[j]) c3++;

        }

    }
```

```
}  
  
if(c1==strlen(s) || c2==strlen(s) || c3==strlen(s))  
    printf("true");  
  
else  
    printf("false");  
  
return 0;  
}
```

sir if i do not get campus placement in final year..
is there opportunity for freshers off campus after btech
because i am from mechanical so

**** little bit fear i always have

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```



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

String s = obj.nextLine();

String r1="qwertyuiop";

String r2="asdfghjkl";

String r3="zxcvbnm";

int i,j,c1=0,c2=0,c3=0;

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

    for(j=0;j<r1.length();j++){

        if(s.charAt(i)==r1.charAt(j)) c1++;

    }

    for(j=0;j<r2.length();j++){

        if(s.charAt(i)==r2.charAt(j)) c2++;

    }

    for(j=0;j<r3.length();j++){

        if(s.charAt(i)==r3.charAt(j)) c3++;

    }

}

System.out.println((c1==s.length()) || (c2==s.length()) || (c3==s.length()));

}
```

python implementation:

```
s=input()
c1,c2,c3=0,0,0
for i in s:
    if i in "qwertyuiop":
        c1=c1+1
    if i in "asdfghjkl":
        c2=c2+1
    if i in "zxcvbnm":
        c3=c3+1
print("true" if (c1==len(s) or c2==len(s) or c3==len(s)) else "false")
```

LBP89

Rotate String

Given two strings s and goal, return true if and only if s can become goal after some number of shifts on s.

A shift on s consists of moving the leftmost character of s to the rightmost position.

For example, if s = "abcde", then it will be "bcdea" after one shift.

abcde

bcdea

cdeab

deabc

eabcd

x and y

abcde

bcdea

abcdeabcde

bcdea

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
char s1[100],s2[100],s3[100]="\0";

int i,j=0,c=0;

scanf("%s",s1);

scanf("%s",s2);

i=0;

while(s1[i]){

    s3[j++]=s1[i++];

}

i=0;

while(s1[i]){

    s3[j++]=s1[i++];

}

j=0;

for(i=0;s3[i];i++){

    if(s3[i]==s2[j]){

        c++;

        j++;

    }

}

printf((c==strlen(s2))?"true":"false");

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s1 = obj.nextLine();
```

```
        String s2 = obj.nextLine();
```

```
        System.out.println((s1+s1).contains(s2));
```

```
    }
```

```
}
```

python implementation:

```
s1=input()
```

```
s2=input()
```

```
print("true" if s2 in s1+s1 else "false")
```

LBP90

Missing Letters

Given a string containing unique letters, return a sorted string with the letters that don't appear in the string.

input -----> a string from the user

con -----> no

output -----> return missing characters

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,a[256]={0};
```

```
    scanf("%s",s);
```

```
for(i=0;s[i];i++){
    a[s[i]]++;
}
for(i=97;i<=122;i++)
{
    if(a[i]==0)
        printf("%c",i);
}
return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
        Scanner obj = new Scanner(System.in);
        String s = obj.nextLine();
        StringBuffer sb = new StringBuffer();
        for(int i='a';i<='z';i++){
```

```
        if(s.indexOf(i)==-1)

            sb.append((char)i);

        }

        System.out.println(sb);

    }

}
```

python implementation:

```
-----

s=input()

for i in range(97,123):

    if chr(i) not in s:

        print(chr(i),end="")
```

LBP91

Replace Letters With Position In Alphabet

Create a function that takes a string and replaces each letter with its appropriate position in the alphabet.

"a" is 1, "b" is 2, "c" is 3, etc, etc.

Note: If any character in the string isn't a letter, ignore it.

input -----> a string from the user

constraint -----> non-empty string

output -----> position of characters separated by space

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i;
```

```
    scanf("%[^\\n]s",s);
```

```
    for(i=0;s[i];i++){
```

```
        if(s[i]>='A' && s[i]<='Z')
```

```
            s[i]=s[i]+32;
```

```
    }
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        if(s[i]>='a' && s[i]<='z')
```

```
        printf("%d ",s[i]-96);  
    }  
  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine().toLowerCase();  
        for(int i=0;i<s.length();i++){  
            if(s.charAt(i)>='a' && s.charAt(i)<='z')  
                System.out.print((s.charAt(i)-96)+" ");  
        }  
    }  
}
```

python implementation:

```
-----  
s=input().lower()  
for i in range(len(s)):  
    if s[i]>='a' and s[i]<='z':  
        print(ord(s[i])-96,end=' ')
```

LBP92

Replace character with it's occurrence

Implement a Program to replace a character with it's occurrence in given string.

input -----> a string and a character from the user.

con -----> non-empty string

output -----> replaced string

c implementation:

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

```
#include <stdlib.h>
```

```
int main() {  
    char s[100],ch;  
    int c=1;  
    scanf("%s",s);  
    scanf("\n%c",&ch);  
    for(int i=0;s[i];i++){  
        if(s[i]==ch)  
            printf("%d",c++);  
        else  
            printf("%c",s[i]);  
    }  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    String s = obj.nextLine();  
    char ch = obj.nextLine().charAt(0);  
    int i,c=1;  
    for(i=0;i<s.length();i++){  
        if(s.charAt(i)==ch)  
            System.out.print(c++);  
        else  
            System.out.print(s.charAt(i));  
    }  
}
```

python implementation:

```
s=input()  
ch=input()  
c=1  
for i in s:  
    if i==ch:  
        print(c,end="")
```

```
c=c+1
```

```
else:
```

```
    print(i,end="")
```

LBP93

first non-repeated character

Program to find first non-repeated character

input----> a non-empty string from the user

con -----> no

output --> non-repeated character

india ---> nda

indian --> da

c implementation:

```
-----
```

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,j,u;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++){
```

```
        u=1;
```

```
        //logic
```

```
        for(j=0;s[j];j++){
```

```
            if(i!=j && s[i]==s[j]){
```

```
                u=0;
```

```
                break;
```

```
            }
```

```
        }
```

```
        if(u==1){
```

```
            printf("%c",s[i]);
```

```
            break;
```

```
        }
```

```
    }
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s=obj.nextLine();
```

```
        int i,u,j;
```

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

```
            u=1;
```

```
            for(j=0;j<s.length();j++){
```

```
                if(i!=j && s.charAt(i)==s.charAt(j)){
```

```
                    u=0;
```

```
                    break;
```

```
                }
```

```
            }
```

```
            if(u==1){
```

```
                System.out.print(s.charAt(i));
```



```
        break;
    }
}
}
```

python implementation:

```
s=input()
for i in s:
    if s.count(i)==1:
        print(i)
        break
```

LBP94

Pangrams

Implement a program to check whether the given string pangram or not.

A pangram is a string that contains all the letters of the English alphabet.

An example of a pangram is "The quick brown fox jumps over the lazy dog"

input ----> a string from the user

con -----> non-empty string

output ---> Yes or No

logic:

a[26]={0}

a[s[i]-97]++;

abc

a[97-97]=a[0]++=1

a[98-97]=a[1]++=1

a[99-97]=a[2]++=1

if a[i]=0 then it is Not Pangram else Yes

c implementation:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

```
int main() {  
    char s[100];  
    int flag=1,i,a[26]={0};  
    scanf("%[^\\n]s",s);  
    for(i=0;s[i];i++)  
        a[s[i]-97]++;  
    for(i=0;i<26;i++)  
    {  
        if(a[i]==0){  
            flag=0;  
            break;  
        }  
    }  
    printf((flag==1)?"Yes":"No");  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    String s = obj.nextLine();  
    boolean p = true;  
    for(int i='a';i<='z';i++){  
        if(s.indexOf(i)<0){  
            p=false;  
            break;  
        }  
    }  
    System.out.println(p?"Yes":"No");  
}
```

python implementation:

```
s=input()  
ss="abcdefghijklmnopqrstuvwxyz"  
flag=True  
for i in ss:  
    if i not in s:  
        flag=False
```

break

```
print("Yes" if flag else "No")
```

LBP95

Print First Letter of each Word

Implement a function/Method to return first character in each word from the given input string.

input-----> a string

con-----> no

output ---> first character in each string

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
char s[100],*p;

scanf("%[^\\n]s",s);

p=strtok(s," ");

while(p!=NULL){

    printf("%c",p[0]);

    p=strtok(NULL," ");

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String[] s = obj.nextLine().split(" ");

        for(String ss:s)

            System.out.print(ss.charAt(0));

    }

}
```

```
}  
  
}
```

python implementation:

```
-----  
  
print(''.join([i[0] for i in input().split()])))
```

LBP96

Number of vowels

Implement a program to return number of vowels present in the given string

input -----> a string from the user
con -----> non-empty string
output -----> return number of vowels

c implementation:

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

```
#include <stdlib.h>
```

```
int main() {  
    char s[100];  
    int i,c=0;  
    scanf("%s",s);  
    for(i=0;s[i];i++){  
        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')  
            c++;  
    }  
    printf("%d",c);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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



```
String s = obj.nextLine();

int i,c=0;

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

    if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' || s.charAt(i)=='o' || s.charAt(i)=='u')

        c++;

}

System.out.println(c);

}
```

python implementation:

```
s=input()

count=0

for i in s:

    if i in "aeiou":

        count+=1

print(count)
```

LBP97

Number of consonants

Implement a program to return number of consonants present in the given string

input -----> a string from the user

con -----> non-empty string

output -----> return number of consonants

c implementation:

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

```
int main() {
    char s[100];
    int i,c=0;
    scanf("%s",s);
    for(i=0;s[i];i++){
        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
            continue;
        else
```

```
        c++;  
  
    }  
  
    printf("%d",c);  
  
    return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int i,c=0;  
  
        for(i=0;i<s.length();i++){  
  
            if(s.charAt(i)=='a' || s.charAt(i)=='e' || s.charAt(i)=='i' || s.charAt(i)=='o' || s.charAt(i)=='u')  
  
                continue;  
  
            else  
  
                c++;  
  
        }  
    }  
}
```

```
}  
  
System.out.println(c);  
  
}  
  
}
```

python implementation:

```
-----  
  
s=input()  
  
count=0  
  
for i in s:  
  
    if i not in "aeiou":  
  
        count+=1  
  
print(count)
```

LBP98

Check only digits

Implement a program to check if a string contains only digits.

input ----> a string from the user

con -----> no

output ---> Yes or No

"123" ----> 3==3 Yes

"abc12" --> 2==5 No

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100];

    int i,c=0;

    scanf("%s",s);

    for(i=0;s[i];i++){

        if(s[i]>='0' && s[i]<='9')

            c++;

    }

    printf((c==strlen(s))?"Yes":"No");

    return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        int i,c=0;
```

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

```
            if(s.charAt(i)>='0' && s.charAt(i)<='9')
```

```
                c++;
```

```
        }
```

```
        System.out.println((c==s.length())?"Yes":"No");
```

```
    }
```

```
}
```

python implementation:

```
print("Yes" if input().isdigit() else "No")
```

LBP99

Capitalize Every word first character

Implement a program to capitalize first letter of each word in a string.

input ----> a string from the user

con -----> non-empty string

output ---> a string with capitalization

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100], *p;
```

```
    scanf("%[^\n]s", s);
```

```
    p = strtok(s, " ");
```

```
    while(p != NULL){
```

```
printf("%c%s ",p[0]-32,p+1);

p=strtok(NULL," ");

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s[] = obj.nextLine().split(" ");

        for(String ss:s){

            System.out.print(ss.substring(0,1).toUpperCase()+ss.substring(1)+" ");

        }

    }

}
```

python implementation:

`print(input().title())`

LBP100

Student Rewarded

You are given a string representing an attendance record for a student.

The record only contains the following three characters:

'A' : Absent.

'L' : Late.

'P' : Present.

A student could be rewarded if his attendance record doesn't contain
more than one 'A' (absent) or
more than two continuous 'L' (late).

You need to return whether the student could be rewarded according to his
attendance record.

input -----> a string from the user

con -----> non empty string

output -----> Yes or No

c1 =====> n(A)=1, LLL not allowed

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100];

    int ac=0,lc=0;

    scanf("%s",s);

    for(int i=0;s[i];i++){

        if(s[i]=='A') ac++;

        if(s[i]=='L' && s[i+1]=='L' && s[i+2]=='L') lc++;

    }

    printf((lc==1 || ac>1)?"No":"Yes");

    return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s=obj.nextLine();
```

```
        int ac=0;
```

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

```
            if(s.charAt(i)=='A')
```

```
                ac++;
```

```
        }
```

```
        if(ac>1 || s.contains("LLL"))
```

```
            System.out.println("No");
```

```
        else
```

```
            System.out.println("Yes");
```

```
    }
```

```
}
```

python implementation:

```
s=input()
```

```
print("No" if s.count("A")>1 or s.count("LLL")!=0 else "Yes")
```

sir i have doubt in strtok ..

sir in while loop at last line

```
p=strtok(NULL," ");
```

sir i am not able to understand this line..

sir if possible plz graphically explain this line

"welcome to python programming"

```
p=strtok(s," ");
```

```
while(p!=NULL){
```

```
    -----
```

```
    -----
```

```
    -----
```

```
    p=strtok(NULL," ");
```

```
}
```

p=welcome

p=to

p=python

p=programming

p=NULL

LBP101

reading and writing an array

Implement a program to read an array element and write on the screen.

input -----> size of the array and array elements

con -----> size<100

output -----> the given array

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();  
        int i;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)
```

```
a[i]=obj.nextInt();  
for(i=0;i<n;i++)  
    System.out.print(a[i]+" ");  
}  
}
```

python implementation:

```
-----  
n=int(input())  
l=[int(i) for i in input().split()]  
for i in l:  
    print(i,end=' ')
```

LBP101 TO LBP200 ==> VERY EFFICIENT IN ARRAYS IN ALL LANGUAGES

Beer ==> A,B,C,D -----> Z

LBP102

sum of all elements in array

Implement a program to read an array elements and print sum of all its elements.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of all elements

c implementation:

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

```
int main() {
    int n,a[100],i,s=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)
    {
        s=s+a[i];
    }
}
```



```
printf("%d",s);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i,s=0;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
        {  
            s=s+a[i];  
        }  
  
        System.out.print(s);
```

```
}  
  
}
```

python implementation:

```
-----  
  
n=int(input())  
print(sum([int(i) for i in input().split()]))
```

LBP103

sum of even numbers in an array

Implement a program to read an array elements and print sum of all even elements.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of all even elements

c implementation:

```
-----
```

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,s=0;
```

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

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

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

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

```
    {
```

```
        if(a[i]%2==0)
```

```
            s=s+a[i];
```

```
    }
```

```
    printf("%d",s);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();  
        int i,s=0;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
        for(i=0;i<n;i++)  
        {  
            if(a[i]%2==0)  
                s=s+a[i];  
        }  
        System.out.print(s);  
    }  
}
```

python implementation:

```
n=int(input())  
print(sum([int(i) for i in input().split() if int(i)%2==0]))
```

LBP104

sum of odd numbers in an array

Implement a program to read an array elements and print sum of all odd elements.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of all odd elements

c implementation:

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

```
int main() {
    int n,a[100],i,s=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
```

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

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i,s=0;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)
```

```
a[i]=obj.nextInt();  
for(i=0;i<n;i++)  
{  
    if(a[i]%2!=0)  
        s=s+a[i];  
}  
System.out.print(s);  
}  
}
```

python implementation:

```
n=int(input())  
print(sum([int(i) for i in input().split() if int(i)%2==0]))
```

LBP105

sum of prime numbers in an array

Implement a program to read an array elements and print sum of all prime elements.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of all prime elements

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int isprime(int n){
```

```
    int i,f=0;
```

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

```
        if(n%i==0)
```

```
            f++;
```

```
    }
```

```
    return f==2;
```

```
}
```

```
int main() {
```

```
    int n,a[100],i,s=0;
```

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

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

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



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

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
    static boolean isprime(int n){  
        int i,f=0;  
        for(i=1;i<=n;i++){  
            if(n%i==0)  
                f++;  
        }  
        return f==2;  
    }  
}
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt();  
    int i,s=0;  
    int a[]=new int[n];  
    for(i=0;i<n;i++)  
        a[i]=obj.nextInt();  
    for(i=0;i<n;i++)  
    {  
        if(isprime(a[i]))  
            s=s+a[i];  
    }  
    System.out.print(s);  
}  
}
```

python implementation:

```
def isprime(n):  
    f=0  
    for i in range(1,n+1):  
        if n%i==0:
```

f=f+1

return f==2

n=int(input())

print(sum([int(i) for i in input().split() if isprime(int(i))]))

LBP106

sum of palindrome numbers in an array

Implement a program to read an array elements and print sum of all palindrome numbers in array.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of all palindrome numbers

logic:

read n and a[n]

s=0

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

```
{  
    if(ispali(a[i]))  
        s=s+a[i];  
}
```

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
int ispali(int n){  
    int r=0,t,d;  
  
    t=n;  
  
    while(n!=0){  
        d=n%10;  
        r=r*10+d;  
        n=n/10;  
    }  
  
    return t==r;  
}  
  
int main() {  
  
    int n,a[100],i,s=0;
```

```
scanf("%d",&n);  
  
for(i=0;i<n;i++)  
    scanf("%d",&a[i]);  
  
for(i=0;i<n;i++)  
{  
    if(ispali(a[i]))  
        s=s+a[i];  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    static boolean ispali(int n){  
  
        int r=0,t=n,d;  
  
        while(n!=0)  
  
        {
```

```
d=n%10;

r=r*10+d;

n=n/10;

}

return t==r;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n=obj.nextInt(),s=0,i;

    int a[]=new int[n];

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

        a[i]=obj.nextInt();

    s=0;

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

        if(ispali(a[i]))

            s=s+a[i];

    }

    System.out.println(s);

}

}
```

python implementation:

1st version:

```
n=int(input())
```

```
L=[i for i in input().split()]
```

```
s=0
```

```
for i in L:
```

```
    if i==i[::-1]:
```

```
        s=s+int(i)
```

```
print(s)
```

2nd version:

```
n=int(input())
```

```
print(sum([int(i) for i in input().split() if i==i[::-1]]))
```

LBP107

sum of strong numbers in an array

Implement a program to read an array elements and print sum of all strong numbers in array.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of all strong numbers

$123=1!+2!+3!=1+2+6=9$ not strong

$145=1!+4!+5!=1+24+120=145$ yes it is strong number

logic:

```
int isstrong(){-----}
```

```
int factorial(){----}
```

read n and a[n]

s=0

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

```
{
```

```
    if(isstrong(a[i]))
```

```
        s=s+a[i];
```

```
}
```

c implementation:

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



```
#include <string.h>

#include <math.h>

#include <stdlib.h>

int factorial(int n){
    if(n==0)
        return 1;
    else
        return n*factorial(n-1);
}

int isstrong(int n){
    int s=0,d,t;
    t=n;
    while(n!=0){
        d=n%10;
        s=s+factorial(d);
        n=n/10;
    }
    return t==s;
}

int main() {
    int n,a[100],i,s=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
```

```
scanf("%d",&a[i]);  
  
for(i=0;i<n;i++)  
{  
    if(isstrong(a[i]))  
        s=s+a[i];  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
    static int factorial(int n)  
    {  
        if(n==0)  
            return 1;  
        else  
            return n*factorial(n-1);  
    }  
}
```

```
static boolean isstrong(int n){  
    int s=0,t=n,d;  
    while(n!=0)  
    {  
        d=n%10;  
        s=s+factorial(d);  
        n=n/10;  
    }  
    return t==s;  
}  
  
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt(),s=0,i;  
    int a[]=new int[n];  
    for(i=0;i<n;i++)  
        a[i]=obj.nextInt();  
    s=0;  
    for(i=0;i<n;i++){  
        if(isstrong(a[i]))  
            s=s+a[i];  
    }  
    System.out.println(s);  
}
```

```
}
```

python implementation:

```
import math
```

```
def isstrong(n):
```

```
    t=n
```

```
    s=0
```

```
    while n!=0:
```

```
        d=n%10
```

```
        s=s+math.factorial(d)
```

```
        n=n//10
```

```
    return s==t
```

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
s=0
```

```
for i in L:
```

```
    if isstrong(i):
```

```
        s=s+i
```

```
print(s)
```

LBP108

sum of elements in an array ending with 3

Implement a program to read an array elements and print sum of elements ending with 3 in array.

input -----> size of the array and array elements

con -----> size<100

output -----> sum of elements ending with 3

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,s=0;
```

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

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

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

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

```
{
```

```
        if(a[i]%10==3)
            s=s+a[i];
    }
    printf("%d",s);
    return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),s=0,i;

        int a[]=new int[n];

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

            a[i]=obj.nextInt();

        s=0;

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

            if(a[i]%10==3)

                s=s+a[i];

        }

    }

}
```

```
}  
  
System.out.println(s);  
  
}  
  
}
```

python implementation:

```
n=int(input())  
print(sum([int(i) for i in input().split() if i.endswith('3')]))
```

LBP109

search for an element in an array

Implement a program to search for an element in an array.

input -----> size, array elements and element to search

con -----> size<100

output -----> search for the given element

logic:

index=-1;

```
for(i=0;i<n;i++){  
    if(key==a[i]){  
        index=i;  
        break;  
    }  
}  
  
print index
```

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
    int n,a[100],index,i,key;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    scanf("%d",&key);  
    index=-1;  
    for(i=0;i<n;i++)
```



```
{  
    if(a[i]==key)  
    {  
        index=i;  
        break;  
    }  
}  
printf("%d",index);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),index=-1,i;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)
```

```
a[i]=obj.nextInt();  
  
int key=obj.nextInt();  
  
for(i=0;i<n;i++)  
{  
    if(key==a[i]){  
        index=i;  
        break;  
    }  
}  
  
System.out.println(index);  
}  
}
```

python implementation:

```
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
key=int(input())  
  
index=-1  
  
for i in range(0,len(L)):  
    if key==L[i]:  
        index=i  
  
        break
```

print(index)

LBP110

sort the elements in an array ASC

Implement a program to sort the given array elements in ASC order.

input -----> size and array elements

con -----> size<100

output ----> sorted array in ASC

5

3, 1, 5, 4, 2 =====> alg =====> 1, 2, 3, 4, 5 or 5, 4, 3, 2, 1

```
for(i=0;i<n;i++){  
    for(j=i+1;j<n;j++){  
        if(a[i]>a[j]) then swap  
    }  
}
```

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
  
    int n,a[100],i,j,t;  
  
    scanf("%d",&n);  
  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
  
    for(i=0;i<n;i++){  
        for(j=i+1;j<n;j++){  
            if(a[i]>a[j]){  
                t=a[i];  
                a[i]=a[j];  
                a[j]=t;  
            }  
        }  
    }  
  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]);  
  
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i;
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        Arrays.sort(a);
```

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

```
            System.out.print(a[i]+" ");
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
for i in L:  
    print(i,end=' ')
```

LBP111

sort the elements in an array DESC

Implement a program to sort the given array elements in DESC order.

input -----> size and array elements

con -----> size<100

output ----> sorted array in DESC

5

3, 1, 5, 4, 2 =====> alg =====> 1, 2, 3, 4, 5 or 5, 4, 3, 2, 1

```
for(i=0;i<n;i++){  
    for(j=i+1;j<n;j++){  
        if(a[i]<a[j]) then swap  
    }  
}
```

```
}
```

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

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

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

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

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

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

```
            if(a[i]<a[j]){
```

```
                t=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=t;
```

```
            }
```

```
        }
```

```
    }
```

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

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),i;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
        Arrays.sort(a);  
        for(i=n-1;i>=0;i--)  
            System.out.print(a[i]+" ");  
    }  
}
```


python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort(reverse=True)  
for i in L:  
    print(i,end=' ')
```

LBP112

binary search

Implement a program to search for an element in an array.

input -----> size, array elements and element to search

con -----> size<100

output -----> search for the given element

logic

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
  
    int n,a[100],i,j,t,key,index=-1,low,high,mid;  
  
    scanf("%d",&n);  
  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
  
    scanf("%d",&key);  
  
    for(i=0;i<n;i++){  
        for(j=i+1;j<n;j++){  
            if(a[i]>a[j]){  
                t=a[i];  
                a[i]=a[j];  
                a[j]=t;  
            }  
        }  
    }  
  
    low=0;  
  
    high=n-1;
```

```
while(low<=high){  
    mid=(low+high)/2;  
    if(key==a[mid])  
    {  
        index=mid;  
        break;  
    }  
    else if(key>a[mid])  
        low=mid+1;  
    else  
        high=mid-1;  
}  
printf("%d",index);  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt(),i;  
    int a[]=new int[n];  
    for(i=0;i<n;i++)  
        a[i]=obj.nextInt();  
    int key=obj.nextInt();  
    Arrays.sort(a);  
    int index=Arrays.binarySearch(a,key);  
    System.out.println((index<0)?-1:index);  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
key=int(input())  
L.sort()  
if key in L:  
    print(L.index(key))  
else:
```

print(-1)

cricket....

Player A -----> 98 Runs -----> 102 Runs ----> A

Player B -----> 89 Runs -----> 89 Runs ----->

Player's A position Sorting Technique -----> formulas

1, 2, 3, 4, 5, 6,

a[0], a[1], a[2], a[3], a[4], a[5],, a[n-1]

min = a[0]

max = a[n-1]

1st min = a[1-1] 1st max = a[n-1]

2nd min = a[2-1] 2nd max = a[n-2]

3rd min = a[3-1] 3rd max = a[n-3]

and so on

LBP113

max element in an array

Implement a program to read array elements and find the max element in an array.

input -----> size and array elements.

con -----> size<100

output -----> return max element

c implementation:

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

```
int main() {
    int n,a[100],i,j,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++){
```

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

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i;
```

```
        int a[]=new int[n];
```

```
for(i=0;i<n;i++)  
    a[i]=obj.nextInt();  
Arrays.sort(a);  
System.out.print(a[n-1]);  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
print(L[n-1])
```

LBP114

min element in an array

Implement a program to read array elements and find the min element in an array.

input -----> size and array elements.

con -----> size<100

output -----> return min element

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

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

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

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

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

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

```
            if(a[i]>a[j]){
```

```
                t=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=t;
```

```
            }
```

```
        }
```

```
    }
```

```
printf("%d ",a[1-1]);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        Arrays.sort(a);  
  
        System.out.print(a[1-1]);  
  
    }  
  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
print(L[1-1])
```

LBP115

diff between largest and smallest element in array

Implement a program to read array elements and find the difference between max and min element in an array.

input -----> size and array elements.

con -----> size<100

output -----> return difference between max and min element.

c implementation:

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i,j,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++){  
        for(j=i+1;j<n;j++){  
            if(a[i]>a[j]){  
                t=a[i];  
                a[i]=a[j];  
                a[j]=t;  
            }  
        }  
    }  
    printf("%d ",a[n-1]-a[1-1]);  
    return 0;  
}
```

java implementation:

import java.io.*;

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),i;

        int a[]=new int[n];

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

            a[i]=obj.nextInt();

        Arrays.sort(a);

        System.out.print(a[n-1]-a[1-1]);

    }

}
```

python implementation:

```
n=int(input())

L=[int(i) for i in input().split()]

L.sort()

print(L[n-1]-L[1-1])
```

LBP116

second largest element in an array

Implement a program to read array elements and find the second max element in an array.

input -----> size and array elements.

con -----> size<100

output -----> return second max element in array

c implementation:

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

```
int main() {
    int n,a[100],i,j,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++){
```

```
for(j=i+1;j<n;j++){  
    if(a[i]>a[j]){  
        t=a[i];  
        a[i]=a[j];  
        a[j]=t;  
    }  
}  
}  
  
printf("%d ",a[n-2]);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i;
```

```
        int a[]=new int[n];
```

```
for(i=0;i<n;i++)  
    a[i]=obj.nextInt();  
Arrays.sort(a);  
System.out.print(a[n-2]);  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
print(L[n-2])
```

LBP117

second smallest element in an array

Implement a program to read array elements and find the second min element in an array.

input -----> size and array elements.

con -----> size<100

output -----> return second min element in array

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

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

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

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

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

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

```
            if(a[i]>a[j]){
```

```
                t=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=t;
```

```
            }
```

```
    }
```

```
}  
  
printf("%d ",a[2-1]);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        Arrays.sort(a);  
  
        System.out.print(a[2-1]);  
  
    }  
  
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
print(L[2-1])
```

LBP118

number of occurrences of an element

Implement a program to find the number of occurrences of the given element.

input -----> size,array element and key element

con -----> size<100

output -----> return number of occurrences

logic:

read array size and array elements,

read key (which is to be searched)

```
for(i=0;i<n;i++){  
    if(element==key){  
        count++;  
    }  
}  
  
print count;
```

c implementation:

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

```
int main() {  
  
    int n,a[100],i,count=0,key;  
  
    scanf("%d",&n);  
  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
  
    scanf("%d",&key);  
  
    count=0;  
  
    for(i=0;i<n;i++)  
  
    {
```

```
    if(key==a[i])  
        count++;  
    }  
    printf("%d",count);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int i,key,n=obj.nextInt(),count;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
        key=obj.nextInt();  
        count=0;  
        for(i=0;i<n;i++){
```

```
        if(key==a[i])  
            count++;  
        }  
        System.out.println(count);  
    }  
}
```

python implementation:

```
-----  
n=int(input())  
l=[int(i) for i in input().split()]  
key=int(input())  
print(l.count(key))
```

From Sandeep Kumar to Everyone 08:53 PM

Can we use count = a.lastIndexOf(key)-a.indexOf(key)?

after sorting

LBP119

inserting element at first position in an array

Implement a program to insert an element into an array at the first position

input -----> size,array elements and element to be inserted

con -----> size<100

output -----> return array after insertion

logic:

```
for(i=n;i>=0;i--)
```

```
    a[i]=a[i-1];
```

```
a[0] ----> element
```

5 1 2 3

x 5 1 2 3

8 5 1 2 3

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int a[100],n,i,e;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    scanf("%d",&e);  
    for(i=n;i>=0;i--)  
        a[i]=a[i-1];  
    a[0]=e;  
    n++;  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]);  
    return 0;  
}
```

java implementation:

array ----> fixed in size

collections ---> growable, type is not req

collections with generics ---> growable but type is fixed

```
import java.io.*;
```



```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),i;

        LinkedList<Integer> ll = new <Integer>LinkedList();

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

            ll.add(obj.nextInt());

        ll.addFirst(obj.nextInt());

        for(Object o:ll.toArray()){

            System.out.print((Integer)o+" ");

        }

    }

}
```

sir toady i had wipro test and i did both coding because of u only ...

The way u taught is amazing sir

thank u sir for such quality content

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.insert(0,int(input()))  
for i in L:  
    print(i,end=' ')
```

LBP120

inserting element at last position in an array

Implement a program to insert an element into an array at the last position

input -----> size,array elements and element to be inserted

con -----> size<100

output -----> return array after insertion

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],n,i,e;
```

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

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

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

```
    scanf("%d",&e);
```

```
    a[n]=e;
```

```
    n++;
```

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

```
        printf("%d ",a[i]);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt(),i;  
    LinkedList<Integer> ll = new <Integer>LinkedList();  
    for(i=0;i<n;i++)  
        ll.add(obj.nextInt());  
    ll.addLast(obj.nextInt());  
    for(Object o:ll.toArray()){  
        System.out.print((Integer)o+" ");  
    }  
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.append(int(input()))  
for i in L:  
    print(i,end=' ')
```

LBP121

delete an element from first location in an array

Implement a program to delete an element from an array at the first position

input -----> size,array elements

con -----> size<100

output -----> return array after deleting from first location

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],n,i;
```

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

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

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

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

```
    {
```

```
        a[i]=a[i+1];
```

```
}  
  
n--;  
  
for(i=0;i<n;i++)  
    printf("%d ",a[i]);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        LinkedList<Integer> ll = new <Integer>LinkedList();  
  
        for(i=0;i<n;i++)  
  
            ll.add(obj.nextInt());  
  
        ll.removeFirst();  
  
        for(Object o:ll.toArray()){  
  
            System.out.print((Integer)o+" ");  

```

```
}  
  
}  
  
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.pop(0)  
for i in L:  
    print(i,end=' ')
```

LBP122

delete an element from last location in an array

Implement a program to delete an element from an array at the last position

input -----> size,array elements

con -----> size<100

output -----> return array after deleting from last location

c implementation:

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

```
int main() {
    int a[100],n,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    n--;
    for(i=0;i<n;i++)
        printf("%d ",a[i]);
    return 0;
}
```

java implementation:

```
import java.io.*;
```



```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),i;

        LinkedList<Integer> ll = new <Integer>LinkedList();

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

            ll.add(obj.nextInt());

        ll.removeLast();

        for(Object o:ll.toArray()){

            System.out.print((Integer)o+" ");

        }

    }

}
```

python implementation:

```
n=int(input())

L=[int(i) for i in input().split()]

L.pop(-1) #L.pop(n-1)

for i in L:
```

```
print(i,end=' ')
```

LBP123

delete an element from an array at the given location

Implement a program to delete an element from an array at the position

input -----> size,array elements and position

con -----> size<100

output -----> return array after deleting from the location

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],n,i,loc;
```

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

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

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i;
```

```
        LinkedList<Integer> ll = new <Integer>LinkedList();
```

```
for(i=0;i<n;i++)  
    ll.add(obj.nextInt());  
  
int loc=obj.nextInt();  
  
ll.remove(loc);//here loc is location not object  
  
for(Object o:ll.toArray()){  
    System.out.print((Integer)o+" ");  
}  
  
}  
  
}
```

python implementation:

```
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
L.pop(int(input()))  
  
for i in L:  
    print(i,end=' ')
```

LBP124

delete an element from an array

Implement a program to delete the given element from an array

input -----> size,array elements and element

con -----> size<100

output -----> return array after deleting

c implementation:

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

```
int main() {
    int n,a[100],i,e,status=-1;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    scanf("%d",&e);
    for(i=0;i<n;i++)
    {
        if(e==a[i])
        {
```

```
        status=i;

        break;

    }

}

for(i=status;i<n;i++)

    a[i]=a[i+1];

n--;

if(status!=-1)

{

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

        printf("%d ",a[i]);

}

else

    printf("-1");

return 0;

}
```

Java Implementation:

```
import java.io.*;

import java.util.*;

public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int i,n=obj.nextInt();  
    LinkedList<Integer> ll = new LinkedList<Integer>();  
    for(i=0;i<n;i++)  
        ll.add(obj.nextInt());  
    int e=obj.nextInt();  
    if(ll.remove(new Integer(e))){  
        for(Object o:ll.toArray())  
            System.out.print((Integer)o+" ");  
    }  
    else  
        System.out.println(-1);  
    }  
}
```

Python Implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
e=int(input())  
if e in L:
```

```
L.remove(e)
```

```
for i in L:
```

```
    print(i,end=' ')
```

```
else:
```

```
    print(-1)
```

LBP125

update an element in an array

Implement a program to update an element in the given array

input -----> size,array elements and element to be updated (old element & new element)

con-----> size<100

output -----> return array after updating

c implementation:

```
-----
```

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

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



```
#include <math.h>

#include <stdlib.h>

int main() {

    int a[100],i,oe,ne,n;

    scanf("%d",&n);

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

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

    scanf("%d",&oe);

    scanf("%d",&ne);

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

        if(a[i]==oe)

            a[i]=ne;

    }

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

        printf("%d ",a[i]);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i,oe,ne;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        oe=obj.nextInt();  
  
        ne=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
        {  
  
            if(a[i]==oe)  
  
                a[i]=ne;  
  
        }  
  
        for(i=0;i<n;i++)  
  
            System.out.print(a[i]+" ");  
  
    }  
}
```

python implementation:

```
n=int(input())
L=[int(i) for i in input().split()]
oe=int(input())
ne=int(input())
for i in range(0,n):
    if L[i]==oe:
        L[i]=ne
for i in L:
    print(i,end=' ')
```

LBP126

update an element in an array

Implement a program to update an element in the given array based on position

input -----> size,array elements and element to be updated and location

con-----> size<100

output -----> return array after updating

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i,ne,loc;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    scanf("%d",&loc);  
    scanf("%d",&ne);  
    a[loc]=ne;  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]);  
    return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int i,n=obj.nextInt(),loc,ne;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        loc=obj.nextInt();  
  
        ne=obj.nextInt();  
  
        a[loc]=ne;  
  
        for(i=0;i<n;i++)  
            System.out.print(a[i]+" ");  
    }  
}
```

python implementation:

```
-----  
  
n=int(input())
```

```
l=[int(i) for i in input().split()]
```

```
loc=int(input())
```

```
ne=int(input())
```

```
l[loc]=ne
```

```
for i in l:
```

```
    print(i,end=' ')
```

LBP127

array reverse

Write a program to reverse the elements present in an array

input -----> size, array elements

con -----> size<100

output -----> return array in reverse

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=n-1;i>=0;i--)  
        printf("%d ",a[i]);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),i;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)
```

```
a[i]=obj.nextInt();  
for(i=n-1;i>=0;i--)  
    System.out.print(a[i]+" ");  
}  
}
```

python implementation:

```
-----  
n=int(input())  
l=[int(i) for i in input().split()]  
for i in l[::-1]:  
    print(i,end=' ')
```

LBP128

increment every element in an array by one unit

Implement a program to increment every element by one unit in array.

input -----> size, array elements

con -----> size<100

output -----> increment each element by one unit

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,a[100],i;

    scanf("%d",&n);

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

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

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

        printf("%d ",a[i]+1);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt(),i;  
    int a[]=new int[n];  
    for(i=0;i<n;i++)  
        a[i]=obj.nextInt();  
    for(i=0;i<n;i++)  
        System.out.print((a[i]+1)+" ");  
    }  
}
```

python implementation:

```
-----  
n=int(input())  
l=[int(i) for i in input().split()]  
for i in l:  
    print(i+1,end=' ')
```

LBP129

number of duplicate elements in array

Implement a program to find the number of duplicate elements present in the given array.

input -----> size, array elements

con -----> size<100

output -----> number of duplicate elements in the array

b[999]={0}

11 97 11 23 97 11

b[11]=3

b[23]=1

b[97]=2

>=2

c implementation:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

```
int main() {  
    int a[100],n,i,b[999]={0},c=0;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
    {  
        b[a[i]]++;  
    }  
    for(i=0;i<999;i++)  
    {  
        if(b[i]>=2)  
            c++;  
    }  
    printf("%d",c);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i,c=0,a[]=new int[n];  
  
        int b[]=new int[999];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
            b[a[i]]++;  
  
        for(i=0;i<999;i++)  
  
        {  
  
            if(b[i]>=2)  
  
                c++;  
  
        }  
  
        System.out.println(c);  
  
    }  
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
d={}
```

```
c=0
```

```
for i in L:
```

```
    d[i]=d.get(i,0)+1
```

```
for i in d.values():
```

```
    if i>=2:
```

```
        c=c+1
```

```
print(c)
```

LBP130

print unique elements in array

Implement a program to find the unique elements present in the given array.

input -----> size, array elements

con -----> size<100

output -----> print unique elements present in the array

6

1 2 3 1 4 2

2 3 1 4 2

3 1 4 2

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j;
```

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

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

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

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

```
    {
```

```
        for(j=i+1;j<n;j++)
```

```
        {
```

```
            if(a[i]==a[j])
```

```
                break;
```

```
        }
```

```
    if(j==n)
```

```
        printf("%d ",a[i]);  
  
    }  
  
    return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int i,n=obj.nextInt();  
  
        HashSet<Integer> h = new HashSet<Integer>();  
  
        for(i=0;i<n;i++)  
  
            h.add(obj.nextInt());  
  
        for(Object o:h.toArray())  
  
            System.out.print((Integer)o+" ");  
  
    }  
  
}
```


python implementation:

```
n=int(input())  
L1=[int(i) for i in input().split()]  
L2=[]  
for i in L1:  
    if i not in L2:  
        L2.append(i)  
for i in L2:  
    print(i,end=' ')
```

LBP131

sort an array of 0s, 1s and 2s

Implement a program to read an array and sort array elements with 0s, 1s and 2s.

input -----> size, array elements

con -----> size<100

output -----> print sorted elements

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

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

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

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

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

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

```
            if(a[i]>a[j]){
```

```
                t=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=t;
```

```
            }
```

```
        }
```

```
    }
```

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

```
        printf("%d ",a[i]);
```

```
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        Arrays.sort(a);  
  
        for(i=0;i<n;i++)  
            System.out.print(a[i]+" ");  
  
    }  
  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
for i in L:  
    print(i,end=' ')
```

LBP132

replace every element with the greatest element on its right side

Implement a program to read an array and replace every element with the greatest element on its right side.

input -----> size, array elements

con -----> size<100

output -----> print updated array elements

1 2 3 4 5

5 2 3 4 5

5 5 3 4 5

5 5 5 4 5

5 5 5 5 5

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,max;
```

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

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

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

```
    for(i=0;i<n-1;i++)
```

```
{
```

```
    max=a[i];
```

```
    //max element from i+1 to n
```

```
    for(int j=i+1;j<n;j++)
```

```
{
```

```
        if(max<a[j])
```

```
            max=a[j];
```

```
}
```

```
    a[i]=max;
```

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

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int i,j,max,n=obj.nextInt();  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n-1;i++)  
        {  
            max=a[i];  
  
            for(j=i+1;j<n;j++)
```

```
{  
    if(max<a[j])  
        max=a[j];  
}  
a[i]=max;  
}  
for(i=0;i<n;i++)  
    System.out.print(a[i]+" ");  
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
for i in range(0,n):  
    m=max(L[i:])  
    L[i]=m  
for i in L:  
    print(i,end=' ')
```

LBP133

sum of two arrays

Implement a program to find the sum of two arrays and display the result array

input -----> size and array elements

con -----> no

output -----> print resultant array

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

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

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

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

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

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

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

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



```
printf("%d ",a[i]+b[i]);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int i,n=obj.nextInt();  
  
        int a[]=new int[n];  
  
        int b[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
            b[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
            System.out.print((a[i]+b[i])+" ");  
  
    }
```

```
}
```

python implementation:

```
n=int(input ())  
l1=[int(i) for i in input ().split()]  
l2=[int(i) for i in input (). split ()]  
l3=[]  
for i in range(n):  
    l3.append(l1[i]+l2[i])  
for x in l3:  
    print (x,end=' ')
```

LBP134

sum of elements available at even index

Implement a program to find the sum of elements available at even index locations in an array.

input ----> size and array elements

con -----> no

output ---> print sum value

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

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

    scanf("%d",&n);

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

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

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

    {

        if(i%2==0)

            s=s+a[i];

    }

    printf("%d",s);

    return 0;

}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i,s=0;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
        {  
            if(i%2==0)  
                s=s+a[i];  
        }  
  
        System.out.print(s);  
    }  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
s=0  
  
for i in range(n):  
    if i%2==0:  
        s=s+L[i]  
  
print(s)
```

LBP135

sum of elements available at odd index

Implement a program to find the sum of elements available at odd index locations in an array.

input ----> size and array elements

con -----> no

output ---> print sum value

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i,s=0;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
    {  
        if(i%2!=0)  
            s=s+a[i];  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i,s=0;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
        {  
  
            if(i%2!=0)  
  
                s=s+a[i];  
  
        }  
  
        System.out.print(s);  
  
    }  
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
s=0
```

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

```
    if i%2!=0:
```

```
        s=s+L[i]
```

```
print(s)
```

LBP136

sum of first and last, second and second last and so on

Implement a program to find the sum of first and last, second and second last and so on in an array.

input -----> size and array elements

con -----> no

output ----> print sum of first and last, second and second last and so on

logic:

i=0

j=n-1


```
while(i<=j)
{
    print a[i]+a[j]
    i++
    j--
}
```

c implementation:

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

```
int main() {
    int i,j,a[100],n;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);

    i=0;
    j=n-1;
    while(i<=j)
    {
```

```
printf("%d ",a[i]+a[j]);  
  
i++;  
  
j--;  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int a[]=new int[n];  
  
        int i,j;  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        i=0;j=n-1;  
  
        while(i<=j)
```

```
{  
    System.out.print((a[i]+a[j])+" ");  
    i++;  
    j--;  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
i=0  
j=n-1  
while i<=j:  
    print(L[i]+L[j],end=' ')  
    i=i+1  
    j=j-1
```

LBP137

print reverse of each number in an array

Implement a program to print reverse of each element in an array

input -----> size and array elements

con -----> no

output ----> print reverse of each element in an array

c implementation:

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

int rev(int n)
{
    int d,r=0;
    while(n!=0)
    {
        d=n%10;
        r=r*10+d;
        n=n/10;
    }
    return r;
}
```

```
int main() {  
    int i,a[100],n;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
        printf("%d ",rev(a[i]));  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
    static int rev(int n)  
    {  
        int r=0,d;  
        while(n!=0)  
        {  
            d=n%10;  
            r=r*10+d;  
        }  
    }  
}
```

```
n=n/10;

}

return r;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n=obj.nextInt(),i,a[]=new int[n];

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

        a[i]=obj.nextInt();

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

        System.out.print(rev(a[i])+" ");

}

}
```

python implementation:

```
-----

n=int(input())

l=[i for i in input().split()]

for i in l:

    print(i[::-1],end=' ')
```

LBP138

number of even and odd elements

Implement a program to find number of even and odd elements in the given array

input -----> size and array elements

con -----> no

output -----> print number of even and odd elements line by line

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,i,a[100],c1=0,c2=0;
```

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

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

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

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

```
{
```

```
if(a[i]%2==0)

    c1++;

else

    c2++;

}

printf("%d\n%d",c1,c2);

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i,c1=0,c2=0;
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```



```
for(i=0;i<n;i++)
{
    if(a[i]%2==0)
        c1++;
    else
        c2++;
}
System.out.println(c1+"\n"+c2);
}
```

python implementation:

```
n=int(input())
l=[int(i) for i in input().split()]
c1=0
c2=0
for i in l:
    if i%2==0:
        c1=c1+1
    else:
        c2=c2+1
print(c1)
```

print(c2)

LBP139

Sort only First half of the elements

Implement a program to sort only first half of the array and keep remaining elements as original.

input -----> size and array elements

con -----> no

output -----> sort only first half of the array

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i,j,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n/2;i++){  
        for(j=i+1;j<n/2;j++){  
            if(a[i]>a[j]){  
                t=a[i];  
                a[i]=a[j];  
                a[j]=t;  
            }  
        }  
    }  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        Arrays.sort(a,0,n/2);  
  
        for(i=0;i<n;i++)  
  
            System.out.print(a[i]+" ");  
  
    }  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
for i in range(0,len(L)//2):  
  
    for j in range(i+1,len(L)//2):  
  
        if L[i]>L[j]:  
  
            L[i],L[j]=L[j],L[i]
```

for i in L:

```
print(i,end=' ')
```

LBP140

Difference between two arrays

Implement a program to find the difference between two arrays

input -----> size and array elements

con -----> no

output -----> print difference between two arrays as third array

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],b[100],i;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
        scanf("%d",&b[i]);  
    for(i=0;i<n;i++)  
        printf("%d ",a[i]-b[i]);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int i,n=obj.nextInt();  
        int a[]=new int[n];
```

```
int b[]=new int[n];

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

    a[i]=obj.nextInt();

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

    b[i]=obj.nextInt();

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

    System.out.print((a[i]-b[i])+" ");

}

}
```

python implementation:

```
n=int(input ())

l1=[int(i) for i in input ().split()]

l2=[int(i) for i in input ().split ()]

l3=[]

for i in range(n):

    l3.append(l1[i]-l2[i])

for x in l3:

    print (x,end=' ')
```

LBP141

rearrange an array in such an order that– smallest, largest, 2nd smallest, 2nd largest and on

Implement a program to rearrange an array in such an order that-
smallest,largest,2nd smallest, 2nd largest and so on.

input -----> size and array elements

con -----> no

output -----> print the elements smallest,largest,2nd smallest,2nd largest and
so on.

logic:

i=0

j=n-1

while(i<=j){

 print a[i] and a[j]

 i++

 j--

c implementation:

#include <stdio.h>


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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

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

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

```
    {
```

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

```
    }
```

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

```
    {
```

```
        for(j=i+1;j<n;j++)
```

```
        {
```

```
            if(a[i]>a[j])
```

```
            {
```

```
                t=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=t;
```

```
            }
```

```
        }
```

```
    }
```

```
i=0;

j=n-1;

while(i<=j)

{

    printf("%d %d ",a[i],a[j]);

    i++;

    j--;

}

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i,j;
```

```
        int a[]=new int[n];
```

```
for(i=0;i<n;i++)  
    a[i]=obj.nextInt();  
  
Arrays.sort(a);  
  
i=0;  
j=n-1;  
  
while(i<=j)  
{  
    System.out.print(a[i]+" "+a[j]+" ");  
  
    i++;  
    j--;  
}  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
L.sort()  
  
i=0  
  
j=n-1  
  
while i<=j:  
  
    print(L[i],L[j],end=' ')
```

i=i+1

j=j-1

LBP142

Array of multiples

Implement a program to create an array with n elements by taking multiples of m.

input -----> m and n

con-----> size of the array must be n

output ----> return an array with n elements which contains multiples of m.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
int n,m,i;

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

for(i=1;i<=m;i++)

{

    printf("%d ",i*n);

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),m=obj.nextInt(),i;

        for(i=1;i<=m;i++)

        {

            System.out.print((i*n)+" ");

        }

    }

}
```

```
}  
  
}
```

python implementation:

```
-----  
  
n=int(input())  
m=int(input())  
for i in range(1,m+1):  
    print(n*i,end=' ')
```

LBP143

Inclusive Array Ranges

Write a function that,

given the start startNum and end endNum values,

return an array containing all the numbers inclusive to that range.

Note:

The numbers in the array are sorted in ascending order.

If startNum is greater than endNum, return an array with the higher value.

input -----> n and m values

con -----> no

output ----> return an array with elements from n to m.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,m,a[100],i,j=0;
```

```
    scanf("%d %d",&n,&m);
```

```
    if(n<m)
```

```
    {
```

```
        for(i=n;i<=m;i++)
```

```
        {
```

```
            a[j++]=i;
```

```
        }
```

```
        for(i=0;i<j;i++)
```

```
            printf("%d ",a[i]);
```

```
    }
```

```
    else
```

```
    printf("%d",n);  
  
    return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),m=obj.nextInt(),i,j=0;  
  
        if(n<m){  
  
            int a[]=new int[m-n+1];  
  
            for(i=n;i<=m;i++)  
  
                a[j++]=i;  
  
            for(i=0;i<j;i++)  
  
                System.out.print(a[i]+" ");  
  
        }  
  
        else  
  
            System.out.println(n);  
    }  
}
```



```
}  
  
}  
  
python implementation:
```

```
-----  
  
n=int(input())  
m=int(input())  
L=[]  
if n<m:  
    for i in range(n,m+1):  
        L.append(i)  
    for i in L:  
        print(i,end=' '  
else:  
    print(n)
```

LBP144

Find the Average of the Letters

Create a function that returns the average of an array composed of letters.

First, find the number of the letter in the alphabet in order to find the average of the array.

A = 1

B = 2

C = 3

D = 4

E = 5

average = total sum of all numbers / number of item in the set

Return the result rounded to two decimal points.

input -----> an array as string

con -----> no

output ----> Return the result rounded to two decimal points.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
char s[100];

int i,n,sum=0;

scanf("%s",s);

n=strlen(s);

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

{

    sum=sum+(s[i]-96);

}

printf("%.2f",sum/(float)n);

return 0;

}
```

java implementation:

```
-----

import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s = obj.nextLine();
```

```
int n=s.length(),i,sum=0;

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

    sum=sum+(s.charAt(i)-96);

System.out.printf("%.2f",(sum/(float)n));

}

}
```

python implementation:

```
-----

s=input()

sum=0

for i in s:

    sum=sum+ord(i)-96

print("%.2f"%(sum/len(s)))
```

LBP145

Eliminate Odd Numbers within an Array

Create a function that takes an array of numbers and returns only the even values.

Note:

Return all even numbers in the order they were given.

All test cases contain valid numbers ranging from 1 to 3000.

input -----> size and an array

con -----> no

output ----> remove all odd numbers and print

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i;
```

```
    scanf("%d",&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]);  
  
    }  
  
    return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
        {  
  
            if(a[i]%2==0)  
  
                System.out.print(a[i]+" ");  
  
        }  
  
    }  
  
}
```

```
}  
  
}  
  
}
```

python implementation:

```
n=int(input())  
l=[int(i) for i in input().split()]  
for i in l:  
    if i%2==0:  
        print(i,end=' ')
```

Note: Saturday and Sunday 7PM to 9PM

LBP146

Positive Count / Negative Sum

Create a function that takes an array of positive and negative numbers.

Return an array where the first element is the count of positive numbers and the second element is the sum of negative numbers.

input -----> size and an array

con -----> If given an empty array, return an empty array and 0 is not positive.

output -----> two space separated int values.

logic:

read array elements from the user

if element is +ve then count++

if element is -ve then sum=sum+a[i]

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,c=0,s=0;
```

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



```
for(i=0;i<n;i++)  
    scanf("%d",&a[i]);  
  
if(n!=0)  
{  
    for(i=0;i<n;i++)  
    {  
        if(a[i]>0)  
            c++;  
        else  
            s=s+a[i];  
    }  
    printf("%d %d",c,s);  
}  
  
else  
    printf(" ");  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n = obj.nextInt();  
  
        int i,a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        int c=0,s=0;  
  
        for(i=0;i<n;i++)  
        {  
            if(a[i]>0)  
                c++;  
  
            else  
                s=s+a[i];  
        }  
  
        if(n!=0)  
            System.out.println(c+" "+s);  
  
        else  
            System.out.println(" ");  
    }  
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
c=0
```

```
s=0
```

```
for i in L:
```

```
    if i>0:
```

```
        c=c+1
```

```
    else:
```

```
        s=s+i
```

```
if n!=0:
```

```
    print(c,s)
```

```
else:
```

```
    print(' ')
```

LBP147

Return the Sum of the Two Smallest Numbers

Create a function that takes an array of numbers and returns the sum of the two lowest positive numbers.

input -----> size and an array

con -----> Dn't count negative numbers

output -----> sum of two smallest positive numbers

logic:

sorted array elements

$a[0], a[1], a[2], a[3], \dots, a[n-1]$

if($a[i] \geq 0$) then $a[i] + a[i+1]$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

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

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

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

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i,a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        Arrays.sort(a);
```

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

```
        {
```

```
            if(a[i]>=0){
```

```
                System.out.println(a[i]+a[i+1]);
```

```
                break;
```

```
            }
```

```
        }
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
L.sort()
```

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

```
    if L[i]>=0:
```

```
        print(L[i]+L[i+1])
```

```
        break
```

LBP148

Retrieve the Last N Elements

Write a function that retrieves the last n elements from an array.

input -----> size, an array and N value

con -----> return 0 if n exceeds size of the array

output -----> last N elements

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,m,a[100],i;

    scanf("%d",&n);

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

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

    scanf("%d",&m);

    if(m<=n)

    {

        for(i=n-m;i<n;i++)

            printf("%d ",a[i]);

    }

    else

        printf("0");

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int i,a[]=new int[n];

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

            a[i]=obj.nextInt();

        int m=obj.nextInt();

        if(m<=n)

        {

            for(i=n-m;i<n;i++)

                System.out.print(a[i]+" ");

        }

        else

            System.out.println(0);

    }

}
```

python implementation:

```
-----  
  
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
m=int(input())  
  
if m<n:  
    for i in L[n-m:]:  
        print(i,end=' ')  
  
else:  
    print(0)  
  
  
str(12345)='12345'[::-1][0:3]
```

LBP149

Mini Peaks

Write a function that returns all the elements in an array
that are strictly greater than their adjacent left and right neighbors.

input -----> size, an array

con-----> Do not count boundary numbers, since they only have one
left/right neighbor.

output -----> an array

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[100],i,n;

    scanf("%d",&n);

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

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

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

    {

        if(a[i]>a[i-1] && a[i]>a[i+1])

            printf("%d ",a[i]);

    }

    return 0;

}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i,a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=1;i<n-1;i++)  
        {  
            if(a[i]>a[i-1] && a[i]>a[i+1])  
                System.out.print(a[i]+" ");  
        }  
    }  
}
```

python implementation:

```
-----  
  
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
for i in range(1,len(L)-1):
```

```
    if L[i]>L[i-1] and L[i]>L[i+1]:
```

```
        print(L[i],end=' ')
```

LBP150

All Numbers In Array Are Prime

Create a function that takes an array of integers and returns true if every number is prime.

Otherwise, return false.

input -----> size and an array

con -----> 1 is not a prime number.

output -----> true or false

c implementation:

```
-----
```

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

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

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

```
#include <stdlib.h>
```

```
int isprime(int n)
{
    int f=0,i;
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
            f++;
    }
    return f==2;
}

int main() {
    int n,a[100],i,c=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    c=0;
    for(i=0;i<n;i++)
    {
        if(isprime(a[i]))
            c++;
    }
    if(c==n)
        printf("true");
}
```

```
else  
  
    printf("false");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    static boolean isprime(int n)  
  
    {  
  
        int f=0,i;  
  
        for(i=1;i<=n;i++)  
  
        {  
  
            if(n%i==0)  
  
                f++;  
  
        }  
  
        return f==2;  
  
    }  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);
```

```
int n=obj.nextInt(),i,c=0;

int a[]=new int[n];

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

    a[i]=obj.nextInt();

c=0;

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

{

    if(isprime(a[i]))

        c++;

}

System.out.println(c==n);

}
```

python implementation:

```
def isprime(n):

    f=0

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

        if n%i==0:

            f=f+1

    return f==2

n=int(input())
```



```
L=[int(i) for i in input().split()]  
  
c=0  
  
for i in L:  
    if isprime(i):  
        c=c+1  
  
print("true" if c==n else "false")
```

LBP151

Sum of adjacent Distances

Write a program to calculate and return sum of distances between the adjacent numbers in an array of +ve integers.

input -----> size and array elements

con -----> no

output -----> an int value

5

10 20 30 40 25

10-20=10

20-30=10

30-40=10

40-25=15

sum=10+10+10+15=45

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],n,i,sum;
```

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

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

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

```
    sum=0;
```

```
    for(i=0;i<n-1;i++)
```

```
        sum=sum+abs(a[i]-a[i+1]);
```

```
printf("%d",sum);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);
```

```
        int n=obj.nextInt();
```

```
        int i,sum=0;
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        for(i=0;i<n-1;i++)
```

```
            sum=sum+Math.abs(a[i]-a[i+1]);
```

```
        System.out.println(sum);
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
sum=0  
for i in range(0,n-1):  
    sum=sum+abs(L[i]-L[i+1])  
print(sum)
```

LBP152

Odd Even Online Game

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 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 -----> size and array elements

con -----> no

output -----> all even numbers and odd numbers.

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
    int i,a[100],n;  
  
    scanf("%d",&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]);  
    }  
  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,n=obj.nextInt();
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

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

```
        {
```

```
            if(a[i]%2==0)
```

```
                System.out.print(a[i]+" ");
```

```
        }
```

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

```
        {
```

```
            if(a[i]%2!=0)
```

```
        System.out.print(a[i]+" ");  
    }  
}  
}
```

python implementation:

```
n=int(input())  
L=[int(i) for i in input().split()]  
for i in L:  
    if i%2==0:  
        print(i,end=' ')  
for i in L:  
    if i%2!=0:  
        print(i,end=' ')
```

LBP153

GARMENTS COMPANY APPAREL

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 i/p consists of an integer N, and A1,A2,...AN representing areas of outlets.

output ----> print an integer representing the number of plots that will be selected for outlets.

c implementation:

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

```
int main() {
    int a[100],i,k,n;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    int count=0;
    for(i=0;i<n;i++)
    {
```



```
for(k=1;k<=a[i];k++)  
{  
    if(k*k==a[i])  
        count++;  
}  
}  
printf("%d",count);  
return 0;  
}
```

java implmentation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        int i,k,count=0;
```

```
        int a[]=new int[n];
```

```
for(i=0;i<n;i++)
    a[i]=obj.nextInt();
for(i=0;i<n;i++)
{
    for(k=1;k<=a[i];k++)
    {
        if(k*k==a[i])
            count++;
    }
}
System.out.println(count);
}
```

python implementation:

```
n=int(input())
L=[int(i) for i in input().split()]
count=0
for i in range(0,n):
    for k in range(1,L[i]+1):
        if k*k==L[i]:
            count=count+1
```

print(count)

LBP154

POOLED CAB SERVICE

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 it self is 0.

The distance for the employees who live to the left side of the company is represented with a negative sign.

The distance for the employees who live to the right side of the company is represented with a positive sign.

the cab will be allotted a range of distance.

The company wishes to find the distance for the employees who live within the particular distance range.

write a algorithm to find the distance for the employees who live within the distance range.

input ----> size of the list N ,SD,ED and an array of distance

output ---> distance within the range else -1

con -----> con

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,a[100],i,x1,x2;

    scanf("%d",&n);

    scanf("%d %d",&x1,&x2);

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

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

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

    {

        if(abs(a[i])>=x1 && abs(a[i])<=x2)

            printf("%d ",a[i]);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),x1=obj.nextInt(),x2=obj.nextInt();
```

```
        int a[]=new int[n];
```

```
        int i;
```

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

```
            a[i]=obj.nextInt();
```

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

```
        {
```

```
            if(Math.abs(a[i])>=x1 && Math.abs(a[i])<=x2)
```

```
                System.out.print(a[i]+" ");
```

```
        }
```

```
    }
```

```
}
```

python implementation:

```
-----  
n=int(input())  
x1,x2=(int(i) for i in input().split())  
L=[int(i) for i in input().split()]  
for i in L:  
    if abs(i)>=x1 and abs(i)<=x2:  
        print(i,end=' ')
```

LBP155

Kth SHORTEST PROCESSING QUEUE

A company wishes to modify the technique by which tasks in the processing queue are executed.

There are N processes with unique ID's 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 ----> array size, k value and array

output ---> kth shortest execution time.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],n,i,j,t,k;
```

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

```
    scanf("%d",&k);
```

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

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

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

```
{
```

```
    for(j=i+1;j<n;j++)
```

```
{
```

```
    if(a[i]>a[j])
```

```
{
```

```
        t=a[i];
```

```
        a[i]=a[j];  
        a[j]=t;  
    }  
}  
  
printf("%d",a[k-1]);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),k=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
    }  
}
```



```
Arrays.sort(a);  
  
System.out.println(a[k-1]);  
  
}  
  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
k=int(input())  
  
L=[int(i) for i in input().split()]  
  
L.sort()  
  
print(L[k-1])
```

LBP156

INDEX FILTERING

Create a function that takes two inputs: idx (an array of integers) and str (a string). The function should return another string with the letters of str at each index in idx in order.

input -----> a string followed by size and an array

constraint -----> output must be in lower case but input may not be.

output -----> a string contained in the specified locations given in the array.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i;
```

```
    char s[100];
```

```
    scanf("%[^\\n]s",s);
```

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

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

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

```
    for(i=0;s[i];i++)
```

```
{
```

```
    if(s[i]>='A' && s[i]<='Z')
```

```
        s[i]=s[i]+32;
```

```
}
```

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

```
{
```

```
        printf("%c",s[a[i]]);
```

```
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int n=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
            System.out.print(s.charAt(a[i]));  
  
    }  
}
```

python implementation:

```
s=input()
```

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

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

```
    print(s[L[i]],end="")
```

LBP157

SEVEN BOOM!

Create a function that takes an array of numbers and return "Boom!" if the digit 7 appears in the array. Otherwise, return "there is no 7 in the array".

input -----> an array from the user

constraint ----> no

output -----> Boom! or "there is no 7 in the array".

c implementation:

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

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

```
#include <math.h>

#include <stdlib.h>

int contains(int n)
{
    while(n!=0)
    {
        if(n%10==7)
            return 1;

        n=n/10;
    }

    return 0;
}

int main() {

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

    scanf("%d",&n);

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

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

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

        if(contains(a[i]))

        {

            flag=1;

            break;

        }

    }

}
```

```
}  
  
}  
  
if(flag==1)  
    printf("Boom!");  
  
else  
    printf("there is no 7 in the array");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {  
    static boolean contains(int n)  
    {  
        while(n!=0)  
        {  
            if(n%10==7)  
            {  
                return true;  
            }  
        }  
    }  
}
```

```
}  
  
n=n/10;  
  
}  
  
return false;  
  
}  
  
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    int n=obj.nextInt();  
  
    int a[]=new int[n];  
  
    int i;  
  
    boolean flag=false;  
  
    for(i=0;i<n;i++)  
  
        a[i]=obj.nextInt();  
  
    for(i=0;i<n;i++)  
  
    {  
  
        if(contains(a[i]))  
  
        {  
  
            flag=true;  
  
            break;  
  
        }  
  
    }  
  
    if(flag==true)  
  
        System.out.println("Boom!");  
  
}
```

else

System.out.println("there is no 7 in the array");

}

}

python implementation:

n=int(input())

L=[int(i) for i in input().split()]

flag=False

for i in L:

if '7' in str(i):

flag=True

break

print('Boom!' if flag else "there is no 7 in the array")

LBP158

Positives and Negatives

Create a function which validates whether a given array alternates between positive and negative numbers.

input -----> an array size and array

con -----> no

output -----> true or false

The things that we learnt from mistake, we never forget it

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],n,i,flag=1;
```

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

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

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

```
    for(i=0;i<n-1;i++)
```

```
{
```

```
    if(a[i]>0 && a[i+1]>0)
```

```
{  
    flag=0;  
    break;  
}  
if(a[i]<0 && a[i+1]<0)  
{  
    flag=0;  
    break;  
}  
}  
if(flag==1)  
    printf("true");  
else  
    printf("false");  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int i,n=obj.nextInt();  
    int a[]=new int[n];  
    boolean b=true;  
    for(i=0;i<n;i++)  
        a[i]=obj.nextInt();  
    for(i=0;i<n-1;i++)  
    {  
        if(a[i]>0 && a[i+1]>0)  
        {  
            b=false;  
            break;  
        }  
        if(a[i]<0 && a[i+1]<0)  
        {  
            b=false;  
            break;  
        }  
    }  
    System.out.println(b);  
}
```

}

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
flag=True
```

```
for i in range(0,n-1):
```

```
    if L[i]<0 and L[i+1]<0:
```

```
        flag=False
```

```
        break
```

```
    if L[i]>0 and L[i+1]>0:
```

```
        flag=False
```

```
        break
```

```
print(str(flag).lower())
```

LBP159

Check if All Values Are True

Write a function that returns true if all parameters are truthy, and false otherwise

input -----> an array size and array

con -----> no

output -----> true or false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,flag=1;
```

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

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

```
    {
```

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

```
    }
```

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

```
    {
```

```
        if(a[i]==0)
```

```
        {
```

```
    flag=0;

    break;

}

}

printf((flag==1)?"true":"false");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),i;

        int a[]=new int[n];

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

            a[i]=obj.nextInt();

        boolean b = true;
```

```
for(i=0;i<n;i++)  
{  
    if(a[i]==0)  
    {  
        b=false;  
        break;  
    }  
}  
System.out.println(b);  
}  
}
```

python implementation:

```
n=int(input())  
l=[int(i) for i in input().split()]  
print(str(0 not in l).lower())
```

LBP160

Shared Digits

Create a function that returns true if each pair of adjacent numbers in an array shares at least one digit and false otherwise.

input -----> array size and array elements

con -----> no

output -----> true or false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int contains(int d,int n)
```

```
{
```

```
    while(n!=0)
```

```
    {
```

```
        if(n%10==d)
```

```
            return 1;
```

```
        n=n/10;
```

```
    }
```

```
    return 0;
```

```
}
```



```
int main() {  
    int n,a[100],i,c,x;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    c=0;  
    for(i=0;i<n-1;i++)  
    {  
        x=a[i];  
        while(x!=0)  
        {  
            if(contains(x%10,a[i+1])==1){  
                c++;  
                break;  
            }  
            x=x/10;  
        }  
    }  
    printf((c==n-1)?"true":"false");  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static boolean contains(int d,int n)
```

```
    {
```

```
        while(n!=0)
```

```
        {
```

```
            if(n%10==d)
```

```
                return true;
```

```
            n=n/10;
```

```
        }
```

```
        return false;
```

```
    }
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),c,x,i;
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        c=0;
```

```
for(i=0;i<n-1;i++)  
{  
    x=a[i];  
    while(x!=0)  
    {  
        if(contains(x%10,a[i+1]))  
        {  
            c++;  
            break;  
        }  
        x=x/10;  
    }  
}  
System.out.println(c==n-1);  
}  
}
```

python implementation:

```
n=int(input())  
l=[int(i) for i in input().split()]  
c=0  
for i in range(n-1):
```

```
for j in str(l[i]):  
    if j in str(l[i+1]):  
        c=c+1  
        break  
print(str(c==n-1).lower())
```

LBP161

Combined Consecutive Sequence

Write a function that returns true if two arrays, when combined, form a consecutive sequence.

A consecutive sequence is a sequence without any gaps in the integers,
e.g. 1, 2, 3, 4, 5 is a consecutive sequence, but 1, 2, 4, 5 is not.

input -----> two array sizes and array elements

con -----> no

output -----> true or false

logic:

read array a

read array b

combine both a and b arrays into c array

sort array c

$c[i]+1==c[i+1] \rightarrow \text{counter}++$

$\text{counter}==\text{length of c then true else false}$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n1,n2,a[100],b[100],c[100],i,j,counter=0;
```

```
    scanf("%d",&n1);
```

```
    for(i=0;i<n1;i++)
```

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

```
    scanf("%d",&n2);
```

```
    for(i=0;i<n2;i++)
```

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

j=0;

for(i=0;i<n1;i++)

    c[j++]=a[i];

for(i=0;i<n2;i++)

    c[j++]=b[i];

for(i=0;i<(n1+n2);i++)

{

    for(j=i+1;j<(n1+n2);j++)

    {

        if(c[i]>c[j])

        {

            int t;

            t=c[i];

            c[i]=c[j];

            c[j]=t;

        }

    }

}

for(i=0;i<(n1+n2);i++)

{

    if(c[i]+1==c[i+1])

        counter++;

}
```

```
}  
  
printf((counter==(n1+n2)-1)?"true":"false");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int i,j,counter=0;  
  
        int n1=obj.nextInt();  
  
        int a[]=new int[n1];  
  
        for(i=0;i<n1;i++)  
            a[i]=obj.nextInt();  
  
        int n2=obj.nextInt();  
  
        int b[]=new int[n2];  
  
        for(i=0;i<n2;i++)  
            b[i]=obj.nextInt();
```

```
j=0;

int c[]=new int[n1+n2];

for(i=0;i<n1;i++)

    c[j++]=a[i];

for(i=0;i<n2;i++)

    c[j++]=b[i];

Arrays.sort(c);

for(i=0;i<(n1+n2)-1;i++)

{

    if(c[i]+1==c[i+1])

        counter++;

}

System.out.println(counter==(n1+n2)-1);

}
```

python implementation:

```
-----

n1=int(input())

l1=[int(i) for i in input().split()]

n2=int(input())

l2=[int(i) for i in input().split()]

l3=l1+l2
```



```
l3.sort()

counter=0

for i in range(0,(n1+n2)-1):

    if l3[i]+1==l3[i+1]:

        counter=counter+1

print(str(counter==(n1+n2)-1).lower())
```

LBP162

Count 5s And Win

Arun is obsessed with primes, especially five.

He considers a number to be luckiest if it has the highest number of five in it.

If two numbers have the same frequency of five,

Arun considers the last occurrence of them to be luckiest, and if there is no five in any number,

the first given number is considered luckiest.

Help him choose the luckiest number.

input -----> array size and elements

con -----> no

output -----> return luckiest number

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int contains(int n){

    int c=0;

    while(n!=0){

        if(n%10==5)

            c++;

        n=n/10;

    }

    return c;

}

int main() {

    int n,a[100],i,c=0,x,element,splc=0;

    scanf("%d",&n);

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

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

    c=0;

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

    {
```

```
x=contains(a[i]);//1
if(c<=x)
{
    c=x;
    element=a[i];
}
if(x==0)
    splc++;
}
if(splc!=n)
    printf("%d",element);
else
    printf("%d",a[0]);
return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;

public class Solution {

    static int contains(int n){
```

```
int c=0;

while(n!=0){

    if(n%10==5)

        c++;

    n=n/10;

}

return c;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n=obj.nextInt();

    int a[]=new int[n];

    int i,c=0,splc=0,element=0,x;

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

        a[i]=obj.nextInt();

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

    {

        x=contains(a[i]);

        if(c<=x)

        {

            c=x;

            element=a[i];

        }

    }
```

```
if(x==0)

    splc++;

}

if(splc!=n)

    System.out.println(element);

else

    System.out.println(a[0]);

}

}
```

python implementation:

```
n=int(input())

l=[int(i) for i in input().split()]

c=0

sc=0

for i in l:

    x=str(i).count('5')

    if c<=x:

        c=x

    element =i

if x==0:

    sc=sc+1
```

```
if sc!=n:  
    print(element)  
  
else:  
    print(l[0])
```

LBP163

Find the Single Number

Write a function that accepts an array of numbers (where each number appears three times except for one which appears only once) and finds that unique number in the array and returns it.

input -----> array size and elements
con -----> no
output -----> return non-repeated number

c implementation:

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

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i,j,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
    {  
        t=0;  
        for(j=0;j<n;j++)  
        {  
            if(i!=j && a[i]==a[j])  
                t++;  
        }  
        if(t==0)  
            printf("%d",a[i]);  
    }  
    return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i,j,t;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
        {  
            t=0;  
  
            for(j=0;j<n;j++)  
            {  
                if(i!=j && a[i]==a[j])  
                    t++;  
            }  
  
            if(t==0)  
                System.out.println(a[i]);  
        }  
    }  
}
```



```
}  
  
}
```

python implementation:

```
-----  
  
n=int(input())  
l=[int(i) for i in input().split()]  
for i in l:  
    t=l.count(i)  
    if t==1:  
        print(i)
```

LBP164

Update Every Element

Implement a progra to update every array element with multiplication of previous and next numbers in array.

input -----> size and array elements

con-----> no

output ----> updated array

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int i,n,a[100];

    scanf("%d",&n);

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

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

    printf("%d ",a[1]);

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

    {

        printf("%d ",a[i-1]*a[i+1]);

    }

    printf("%d",a[n-2]);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int i,n=obj.nextInt();

        int a[]=new int[n];

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

            a[i]=obj.nextInt();

        System.out.print(a[1]+" ");

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

            System.out.print((a[i-1]*a[i+1])+" ");

        System.out.print(a[n-2]);

    }

}
```

python implementation:

```
-----

n=int(input())

l=[int(i) for i in input().split()]

print(l[1],end=' ')
```

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

```
    print(l[i-1]*l[i+1],end=' ')
```

```
print(l[n-2])
```

LBP165

Third Largest and Second smallest

Given an integer array and an integer N denoting the array length as input.

your task is to return the sum of third largest and second minimum elements of the array.

input -----> array size and array elements

con -----> no

output -----> an int value

1st min: a[1-1]

1st max: a[n-1]

2nd min: a[2-1]

2nd max: a[n-2]

3rd min: a[3-1]

3rd max: a[n-3]

formula: a[n-3]+a[2-1]

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
  
int main() {  
    int n,a[100],i,j,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
    {  
        for(j=i+1;j<n;j++){  
            if(a[i]>a[j]){  
                t=a[i];  
                a[i]=a[j];  
                a[j]=t;  
            }  
        }  
    }  
  
    printf("%d",a[n-3]+a[2-1]);  
  
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,n=obj.nextInt();
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        Arrays.sort(a);
```

```
        System.out.println(a[n-3]+a[2-1]);
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())  
l=[int(i) for i in input().split()]  
l.sort()  
print(l[n-3]+l[2-1])
```

Sushant Query:

sir in my exam, I faced a question.

To find day - when date, month and year is given

Sir if possible plz discuss that

output: Day

LBP166

Sales Report

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 -----> space separated integers N and M.

con -----> no

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

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int m,n,a[10][10],i,j,max;
```

```
    scanf("%d %d",&n,&m);
```

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

```
    {
```

```
        for(j=0;j<m;j++){
```

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

```
        }
```

```
    }
```

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



```
max=a[i][0];
for(j=0;j<m;j++){
    if(a[i][j]>max){
        max=a[i][j];
    }
}
printf("%d ",max);
}
return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),m=obj.nextInt(),i,j,max;
```

```
        int a[][]=new int[n][m];
```

```
for(i=0;i<n;i++)
{
    for(j=0;j<m;j++){
        a[i][j]=obj.nextInt();
    }
}

for(i=0;i<n;i++){
    max=a[i][0];
    for(j=0;j<m;j++){
        if(a[i][j]>max){
            max=a[i][j];
        }
    }
    System.out.print(max+" ");
}
}
```

python implementation:

```
n,m=(int(i) for i in input().split())
```

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

```
    L=[int(i) for i in input().split()]
```

```
print(max(L),end=' ')
```

LBP167

Online Game

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 -----> an integer num, representing the number displayed on the screen.

con-----> no

output -----> the count of trailing zeros in the factorial of the given number.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int count(int n){
```

```
    int c=0;
```

```
while(n!=0){  
    if(n%10==0)  
        c++;  
    else  
        break;  
    n=n/10;  
}  
return c;  
}  
  
int main() {  
    int n,i,f=1;  
    scanf("%d",&n);  
    for(i=1;i<=n;i++)  
        f=f*i;  
    printf("%d",count(f));  
    return 0;  
}
```

Java Implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
    static int count(long n){  
        int c=0;  
        while(n!=0){  
            if(n%10==0)  
                c++;  
            else  
                break;  
            n=n/10;  
        }  
        return c;  
    }  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();  
        long i,f=1;  
        for(i=1;i<=n;i++)  
            f=f*i;  
        System.out.println(count(f));  
    }  
}
```

python implementation:

```
-----  
  
import math  
  
def count(n):  
    c=0  
  
    while n!=0:  
        if n%10==0:  
            c=c+1  
  
        else:  
            break  
  
        n=n//10  
  
    return c  
  
n=int(input())  
print(count(math.factorial(n)))
```

LBP168

Array paliandrome

Implement a program to check whether an array is paliandrome or not.

input -----> Array size N and Array Elements

con -----> no

output ----> true or false

121 131 141 151 -----> false

121 131 131 121 -----> true

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,flag,low,high;
```

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

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

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

```
    flag=1;
```

```
    low=0;
```

```
    high=n-1;
```

```
    while(low<=high){
```

```
        if(a[low]!=a[high]){
```

```
            flag=0;
```

```
        break;

    }

    low++;

    high--;

}

printf((flag==1)?"true":"false");

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),i,low,high,flag=1;
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```



```
low=0;

high=n-1;

while(low<=high){

    if(a[low]!=a[high]){

        flag=0;

        break;

    }

    low++;

    high--;

}

System.out.println(flag==1);

}
```

python implementation:

```
-----

n=int(input())

L=[int(i) for i in input().split()]

print("true" if L==L[::-1] else "false")
```

LBP169

Array to Matrix

Implement a program to convert an array into matrix.

input -----> array size and elements

con -----> element count should be 1,4,9,16,25 and so on

output ----> matrix

c implementation:

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

```
int main() {
    int n,m,i,j,a[20],k;
    scanf("%d",&m);
    for(i=0;i<m;i++)
        scanf("%d",&a[i]);
    n=sqrt(m);
    k=0;
    for(i=0;i<n;i++){
        for(j=0;j<n;j++){
```

```
        printf("%d ",a[k++]);  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),m=(int)Math.sqrt(n),i,j,k=0;  
        int a[]=new int[n];  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
        for(i=0;i<m;i++)  
        {  
            for(j=0;j<m;j++)
```

```
{  
    System.out.print(a[k++]+" ");  
}  
System.out.println();  
}  
}
```

python implementation:

```
-----  
import math  
n=int(input())  
m=math.isqrt(n)  
k=0  
l=[int(i) for i in input().split()]  
for i in range(m):  
    for j in range(m):  
        print(l[k],end=' ')  
        k=k+1  
    print()
```

LBP170

Matrix to Array

Implement a program to convert the given matrix into array

input -----> matrix of size mxn and elements

con -----> one D array is required

output ----> one-D array should be printed on screen

c implementation:

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

```
int main() {
    int n,m,a[10][10],i,j;
    scanf("%d %d",&n,&m);
    for(i=0;i<n;i++)
    {
        for(j=0;j<m;j++){
            scanf("%d",&a[i][j]);
        }
    }
```

```
}  
  
for(i=0;i<n;i++)  
{  
    for(j=0;j<m;j++){  
        printf("%d ",a[i][j]);  
    }  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt(),m=obj.nextInt(),i,j;
```

```
        int a[][]=new int[n][m];
```

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

```
for(j=0;j<m;j++){  
    a[i][j]=obj.nextInt();  
}  
}  
  
for(i=0;i<n;i++){  
    for(j=0;j<m;j++){  
        System.out.print(a[i][j]+" ");  
    }  
}  
}
```

python implementation:

```
n,m=(int(i) for i in input().split())  
for i in range(n):  
    L=[int(i) for i in input().split()]  
    for j in L:  
        print(j,end=' ')
```

LBP171

Word Key

One programming language has the following keywords that cannot be used as identifiers.

break,case,continue,default,defer,else,for,func,goto,if,map,range,return,struct
,type,var

write a program to find if the given word is a keyword or not.

input -----> string from the user

con -----> con

output -----> true or false

break ----> true

class ----> false

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```



```
char
s[100],*p,ss[]="break,case,continue,default,defer,else,for,func,goto,if,map,ran
ge,return,struct,type,var";

scanf("%s",s);

int flag=0;

p=strtok(ss,"");

while(p!=NULL){

    if(strcmp(p,s)==0)

    {

        flag=1;

        break;

    }

    p=strtok(NULL,"");

}

printf((flag==1)?"true":"false");

return 0;

}
```

Java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        String  
ss[]={ "break", "case", "continue", "default", "defer", "else", "for", "func", "goto", "if",  
        , "map", "range", "return", "struct", "type", "var" };  
  
        boolean flag=false;  
  
        for(String sss:ss){  
  
            if(sss.equals(s)){  
  
                flag=true;  
  
                break;  
  
            }  
  
        }  
  
        System.out.println(flag);  
  
    }  
}
```

python implementation:

```
L=["break", "case", "continue", "default", "defer", "else", "for", "func", "goto", "if", "  
map", "range", "return", "struct", "type", "var"]
```

```
s=input()
print("true" if s in L else "false")
```

LBP172

Oddly Even

Given a maximum of 100 digit numbers as input, find the difference between the sum of odd and even position digits.

input -----> a number from the user

con -----> no

output -----> an integer

45712

01234

$4+7+2=13$

$5+1=6$

$13-6=7$

45712 ==> 21754

4,5,7,1,2

0 1 2 3 4

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int rev(int n){
```

```
    int r=0,d;
```

```
    while(n!=0){
```

```
        d=n%10;
```

```
        r=r*10+d;
```

```
        n=n/10;
```

```
    }
```

```
    return r;
```

```
}
```

```
int main() {
```

```
int n,i,d,a[100],se,so;

scanf("%d",&n);

n=rev(n);

i=0;

while(n!=0){

    d=n%10;

    a[i++]=d;

    n=n/10;

}

int len=i;

se=0;

so=0;

for(i=0;i<len;i++)

{

    if(i%2==0)

        se=se+a[i];

    else

        so=so+a[i];

}

printf("%d",so-se);

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static int reverse(int n){
```

```
        int r=0,d;
```

```
        while(n!=0){
```

```
            d=n%10;
```

```
            r=r*10+d;
```

```
            n=n/10;
```

```
        }
```

```
        return r;
```

```
    }
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        n=reverse(n);
```

```
        int a[]=new int[100],i=0,d;
```

```
        while(n!=0){
```

```
            d=n%10;
```

```
            a[i++]=d;
```

```
n=n/10;

}

int len=i,se=0,so=0;

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

    if(i%2==0)

        se=se+a[i];

    else

        so=so+a[i];

}

System.out.println(so-se);

}

}
```

python implementation:

```
n=input()
n=int(n[::-1])
L=[]
while n!=0:
    L.append(n%10)
    n=n//10
index=0
se=0
```

```
so=0  
  
while index<len(L):  
    if index%2==0:  
        se=se+L[index]  
    else:  
        so=so+L[index]  
    index=index+1  
  
print(so-se)
```

LBP173

Sweet Seventeen

Given a maximum of four digit to the base
17(10=>A,11=>B,12=>C,13=>D,14=>E,15=>F,16=>G) as input, output its
decimal value.

input -----> a string value

con-----> no

output -----> an integer value

$ABC \implies Cx17^0 + Bx17^1 + Cx17^2$

==> Cx1+Bx17+AX289

==> 12+11x17+10x289

==> 12+187+2890

==> 3089

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int dec=0,len,i,t;
```

```
    scanf("%s",s);//ABC
```

```
    len=strlen(s);//3
```

```
    len--;//2
```

```
    for(i=0;s[i];i++){
```

```
        if(s[i]>='0' && s[i]<='9')
```

```
            t=s[i]-48;
```

```
        else if(s[i]>='A'&&s[i]<='G')
```

```
t=s[i]-65+10;

else if(s[i]>='a'&& s[i]<='g')

    t=s[i]-97+10;

dec = dec + t*pow(17,len);

len--;

}

printf("%d",dec);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        System.out.println(Integer.valueOf(obj.nextLine(),17));

    }

}
```

python implementation:

```
print(int(input()),17))
```

LBP174

BeautifyMe

The cosmetic company "BeautyMe" wishes to know the alphabetic product code from the product barcode. The barcode of the product is a numeric value and the alphabetic product is a string value tagged 'a-j'. The alphabetic range 'a-j' represents the numeric range '0-9'. To produce the alphabetic product code, each digit in the numeric barcode is replaced by the corresponding matching letters.

Write an algorithm to display the alphabetic product code from the numeric barcodes.

input -----> an integer value

con -----> no

output -----> a character

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100];

    int i;

    scanf("%s",s);

    for(i=0;s[i];i++)

    {

        printf("%d",s[i]-97);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        for(int i=0;i<s.length();i++)  
  
            System.out.print((int)s.charAt(i)-97);  
  
    }  
}
```

python implementation:

```
-----  
  
s=input()  
  
for i in s:  
  
    print(ord(i)-97,end=")
```

LBP175

Print Prime Numbers

Implement a program to read a number and print prime numbers upto n separated by commas.

input ----> a number from the user

con -----> no

output ---> comma separated prime numbers

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int isprime(int n){
```

```
    int f=0,i;
```

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

```
    {
```

```
        if(n%i==0)
```

```
            f++;
```

```
    }
```

```
    return f==2;
```

```
}
```

```
int main() {
```

```
int n,i;

scanf("%d",&n);

for(i=2;i<=n;i++)

{

    if(isprime(i))

        printf("%d ",i);

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    static boolean isprime(int n){

        int f=0,i;

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

        {

            if(n%i==0)

                f++;

        }

    }

}
```

```
}  
  
return f==2;  
  
}  
  
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    int n=obj.nextInt(),i;  
  
    for(i=2;i<=n;i++)  
  
    {  
  
        if(isprime(i))  
  
            System.out.print(i+" ");  
  
    }  
  
}  
  
}
```

python implementation:

```
def isprime(n):  
  
    f=0  
  
    for i in range(1,n+1):  
  
        if n%i==0:  
  
            f=f+1  
  
    return f==2
```



```
n=int(input())  
for i in range(2,n+1):  
    if isprime(i):  
        print(i,end=' ')
```

LBP176

GCD of two numbers

Implement a program to read two integers values and return GCD of two numbers.

input ----> two space seperated integers.

con -----> no

output ----> GCD of two numbers.

2 3 ----> 1 ----> 1

2 4 ----> 1, 2 --> 2

c implemetation:

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

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

```
#include <math.h>

#include <stdlib.h>

int main() {

    int n1,n2,i,n;

    scanf("%d %d",&n1,&n2);

    for(i=1;i<=n1 && i<=n2;i++)

    {

        if(n1%i==0 && n2%i==0)

            n=i;

    }

    printf("%d",n);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n1=obj.nextInt(),n2=obj.nextInt(),i,n=0;  
    for(i=1;i<=n1&& i<=n2;i++)  
    {  
        if(n1%i==0 && n2%i==0)  
            n=i;  
    }  
    System.out.println(n);  
}  
}
```

python implementation:

```
import math  
n1,n2=(int(i) for i in input().split())  
print(math.gcd(n1,n2))
```

LBP177

secret information

A spy wants to send some secret information to the government. As the data is very important, he encrypts the message by encoding by adding some extra characters. the original message has only upper case letters and numbers, while the extra characters are madeup of lowercase letters and special characters. Can you help the receiver in designing a program that accepts the message and returns the secret information.

input -----> a string from the user

con -----> no

output -----> original message

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    scanf("%s",s);
```

```
    for(int i=0;s[i];i++)
```

```
    {
```

```
        if((s[i]>='A'&& s[i]<='Z') || (s[i]>='0'&& s[i]<='9'))
```

```
            printf("%c",s[i]);
```

```
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        for(int i=0;i<s.length();i++)  
        {  
            if((s.charAt(i)>='A' &&  
s.charAt(i)<='Z') || (s.charAt(i)>='0'&&s.charAt(i)<='9'))  
                System.out.print(s.charAt(i));  
        }  
    }  
}
```

python implementation:

```
-----  
s=input()  
for i in s:  
    if i.isupper() or i.isdigit():  
        print(i,end="")
```

LBP178

flight

amir is travelling to mumbai, but this time he is taking flight. His brother has already told him about luggage weight limits but forgot it. Now he is taking with him 3 trolley bags.

As per the current airlines which amir will fly. has below weight limits.

There can be at max 2 check-in and 1 cabin luggage. Check-in has total limit of L1 and Cabin has limit of L2.

Now, amir has 3 luggage has weights as W1 and W2 and W3 respectively. Now he should be smart enough to make sure that he can travel with all the 3 luggage without paying extra charge.

Find out whether amir can take all of his luggage without any extra charges or not. If all good and no extra changes were paid, output "Yes" otherwise "No".

input -----> integers W1,W2,W3 and L1,L2

con -----> no

output -----> Yes or No

W1,W2,W3 and L1,L2

$W1+W2+W3 \leq L1+L2$ then Yes else No

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int w1,w2,w3,l1,l2;
```

```
    scanf("%d %d %d %d %d",&w1,&w2,&w3,&l1,&l2);
```

```
    if(w1+w2+w3<=l1+l2)
```

```
        printf("Yes");
```

```
else  
  
    printf("No");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int w1=obj.nextInt();  
  
        int w2=obj.nextInt();  
  
        int w3=obj.nextInt();  
  
        int l1=obj.nextInt();  
  
        int l2=obj.nextInt();  
  
        if(w1+w2+w3<=l1+l2)  
  
            System.out.println("Yes");  
  
        else  
  
            System.out.println("No");  
  
    }  
}
```



```
}  
  
}
```

python implementation:

```
-----  
  
w1,w2,w3,l1,l2=(int(i) for i in input().split())  
  
if w1+w2+w3<=l1+l2:  
    print("Yes")  
  
else:  
    print("No")
```

LBP179

arrangement

given an array of size n, write a function to rearrange the numbers of the array in such that even and odd numbers are arranged alternatively in increasing order.

input -----> array size n and elements

con -----> no

output ----> updated array

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,i,j,k,a[100],b[100],c[100],index,t;
```

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

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

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

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

```
{
```

```
    for(j=i+1;j<n;j++)
```

```
{
```

```
    if(a[i]>a[j]){
```

```
        t=a[i];
```

```
        a[i]=a[j];
```

```
        a[j]=t;
```

```
    }
```

```
}
```

```
}
```

```
j=0;k=0;

for(i=0;i<n;i++)
{
    if(a[i]%2==0)
        b[j++]=a[i];
    else
        c[k++]=a[i];
}

index=0;

while(index<n/2)
{
    printf("%d %d ",b[index],c[index]);

    index++;
}

return 0;
}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int a[]=new int[n];

        int b[]=new int[n/2];

        int c[]=new int[n/2];

        int i,index=0,j=0,k=0;

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

            a[i]=obj.nextInt();

        Arrays.sort(a);

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

        {

            if(a[i]%2==0)

                b[j++]=a[i];

            else

                c[k++]=a[i];

        }

        index=0;

        while(index<n/2)

        {

            System.out.print(b[index]+" "+c[index]+" ");
```

```
        index++;  
    }  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L1=[int(i) for i in input().split()]  
L2=[]  
L3=[]  
L1.sort()  
for i in L1:  
    if i%2==0:  
        L2.append(i)  
    else:  
        L3.append(i)  
index=0  
while index<n//2:  
    print(L2[index],L3[index],end=' ')  
    index=index+1
```

LBP180

parity bits

Michael wants to check the parity of the given number. To find the parity, follow below steps.

1. convert decimal number into binary number.
2. count the number of 1's and 0's in the binary representation.

if it contains odd number of 1-bits, then it is "odd parity" and if contains even number of 1-bits then "even parity" Write a program to validate the given number belongs to odd parity or even parity.

input -----> a number from the user.

con -----> no

output -----> odd parity or even parity.

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {  
    int n,c=0;  
    scanf("%d",&n);  
    while(n!=0)  
    {  
        if(n%2==1)  
            c++;  
        n=n/2;  
    }  
    printf((c%2==0)?"even":"odd");  
    return 0;  
}
```

java implementation:

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

```
int n=obj.nextInt();

int c=0;

while(n!=0)

{

    if(n%2==1)

        c++;

    n=n/2;

}

System.out.println((c%2==0)?"even":"odd");

}
```

python implementation:

```
print('even' if bin(int(input()))[2:].count("1")%2==0 else 'odd')
```

LBP181

second non-repeating character

sofia loved to play with the programs unfortunately. she got stuck in one of the questions. she thought to take help of james. james asked her the problem statement. The problem statement was.

"for the given string S of length N consist of stream of character not in predefined order. Write a program to find second non-repeating character in a string S. Write a program to help sofia and james for the given problem statements.

input -----> string from the user

con -----> no

outptu -----> single character

india ---> nda ----> n

indian --> da -----> d

india ---> nda ----> d

indian --> da -----> a

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
char s[100];

int i,j,u;

int c=0;

scanf("%s",s);

for(i=0;s[i];i++){

    u=1;

    //logic

    for(j=0;s[j];j++){

        if(i!=j && s[i]==s[j]){

            u=0;

            break;

        }

    }

    if(u==1){

        c++;

        if(c==2)

            printf("%c",s[i]);

    }

}

return 0;

}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        String s=obj.nextLine();  
  
        int i,u,j;  
  
        int c=0;  
  
        for(i=0;i<s.length();i++){  
            u=1;  
  
            for(j=0;j<s.length();j++){  
                if(i!=j && s.charAt(i)==s.charAt(j)){  
                    u=0;  
  
                    break;  
                }  
            }  
  
            if(u==1){  
                c++;  
  
                if(c==2)  
                {
```

```
        System.out.print(s.charAt(i));  
  
        break;  
  
    }  
  
    }  
  
    }  
  
    }  
  
}
```

python implementation:

```
s=input()  
L=[]  
  
for i in s:  
    if s.count(i)==1:  
        L.append(i)  
  
print(L[1])
```

LBP182

absolute difference between prime numbers

You are given an array of integers, N of size array length.

Find the absolute difference (i.e. $\text{diff} \geq 0$) between the largest and smallest prime numbers.

Note:

1. if there are 1 or fewer prime numbers in array return 0.
2. the array may contain +ve and -ve numbers, ignore -ve numbers.
3. 1 and 0 are not prime.

input -----> array size and array elements

con -----> no

output -----> absolute diff

prime number or not

sort the elements

min=999 =====> min=

max=-1 =====> max=

max!=-1 and min!=999

max-min

else print 0

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int isprime(int n)
{
    int i;
    int f=0;
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
            f++;
    }
    return f==2;
}

int main() {
    int n;
    int a[100];
    int min=999,max=-1;
    int i,j,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
```

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

min=999;
for(i=0;i<n;i++)
{
    if(isprime(a[i]))
    {
        if(min>a[i]){
            min=a[i];
        }
    }
}

max=-1;
```

```
for(i=0;i<n;i++)  
{  
    if(isprime(a[i]))  
    {  
        if(max<a[i]){  
            max=a[i];  
        }  
    }  
}  
  
if(max!=-1 && min!=999)  
    printf("%d",max-min);  
else  
    printf("0");  
return 0;  
}
```

RRR movie

promotional song ==> MNC company names

small edit

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static boolean isprime(int n)
```

```
    {
```

```
        int i;
```

```
        int f=0;
```

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

```
        {
```

```
            if(n%i==0)
```

```
                f++;
```

```
        }
```

```
        return f==2;
```

```
    }
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        int i;
```

```
int a[]=new int[n];  
for(i=0;i<n;i++)  
    a[i]=obj.nextInt();  
Arrays.sort(a);  
int max=-1,min=999;  
for(i=0;i<n;i++)  
{  
    if(isprime(a[i]))  
    {  
        if(min>a[i])  
            min=a[i];  
        if(max<a[i])  
            max=a[i];  
    }  
}  
if(max!=-1 && min!=999)  
    System.out.println(max-min);  
else  
    System.out.println(0);  
}  
}
```

python implementation:

def isprime(n):

 f=0

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

 if n%i==0:

 f=f+1

 return f==2

n=int(input())

L=[int(i) for i in input().split()]

L.sort()

L1=[]

for i in L:

 if isprime(i):

 L1.append(i)

L1.sort()

if len(L1)>=2:

 print(abs(L1[0]-L1[len(L1)-1]))

else:

 print(0)

LBP183

product with successor

Given an integer N and integer array A as the input, where N denotes the length of A

write a program to find an integer the sum of all these product with successors.

input ----> array size and elements

con-----> no

output ---> an int value

c implementation:

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

```
int main() {
    int n;
    int i;
    int a[100];
    int sum=0;
```

```
scanf("%d",&n);  
  
for(i=0;i<n;i++)  
    scanf("%d",&a[i]);  
  
for(i=0;i<n;i++)  
    sum=sum+(a[i]*(a[i]+1));  
  
printf("%d",sum);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int a[]=new int[n];  
  
        int i;  
  
        int sum=0;  
  
        for(i=0;i<n;i++)
```

```
a[i]=obj.nextInt();  
  
for(i=0;i<n;i++)  
  
    sum=sum+(a[i]*(a[i]+1));  
  
System.out.println(sum);  
  
}  
  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
s=0  
  
for i in L:  
  
    s=s+(i*(i+1))  
  
print(s)
```

LBP184

pre-sorted integers in array

You are given an array of integers, a of size n, Your task is to find the number of elements whose positions will remain unchanged after sorted in ascending order.

input ----> array size and elements

con -----> no

output ---> unchanged count

c implementation:

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

```
int main() {
    int n;
    int a[100],b[100];
    int i,c=0,j,t;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    for(i=0;i<n;i++)
        b[i]=a[i];
    for(i=0;i<n;i++)
    {
        for(j=i+1;j<n;j++)
```

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

java implementation:

```
import java.io.*;  
  
import java.util.*;
```



```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int i,n=obj.nextInt();  
  
        int a[]=new int[n];  
  
        int b[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
            b[i]=a[i];  
  
        Arrays.sort(a);  
  
        int c=0;  
  
        for(i=0;i<n;i++)  
        {  
            if(a[i]==b[i])  
                c++;  
        }  
  
        System.out.println(c);  
    }  
}
```

python implementation:

```
-----  
n=int(input())  
L1=[int(i) for i in input().split()]  
L2=L1.copy()  
L1.sort()  
c=0  
for i in range(n):  
    if L1[i]==L2[i]:  
        c=c+1  
print(c)
```

LBP185

savings

There are 3 friends named Ronaldo,Messi,Rooney who worked at a compnay.

Given thier monthly salaries and monthly expendictures, returns the highest saving amoung them.

input -----> single line with 6 space seperated integers.

con -----> no

output -----> highest saving amoung the 3 of them

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a1,a2,b1,b2,c1,c2,a,b,c;

    scanf("%d %d %d %d %d %d",&a1,&a2,&b1,&b2,&c1,&c2);

    a=a1-a2;

    b=b1-b2;

    c=c1-c2;

    printf("%d",(a>b&&a>c)?a:(b>c?b:c));

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int a1=obj.nextInt();  
    int a2=obj.nextInt();  
    int b1=obj.nextInt();  
    int b2=obj.nextInt();  
    int c1=obj.nextInt();  
    int c2=obj.nextInt();  
    int a=a1-a2;  
    int b=b1-b2;  
    int c=c1-c2;  
    System.out.println((a>b&& a>c)?a:(b>c?b:c));  
}  
}
```

python implementation:

```
-----  
a1,a2,b1,b2,c1,c2=(int(i) for i in input().split())  
print(max(a1-a2,b1-b2,c1-c2))
```

LBP186

half ascending and half descending

You are given a list of integers of size N,

write an algorithm to sort the first K elements of the list in ascending order and the remaining elements in descending order.

input -----> an array size and elements

con -----> no

output ----> updated array

5

5 1 2 4 3

1 2 3 4 5

1 2 5 4 3

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int a[100];  
    int n;  
    int i,j,t;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
    {  
        for(j=i+1;j<n;j++)  
        {  
            if(a[i]>a[j])  
            {  
                t=a[i];  
                a[i]=a[j];  
                a[j]=t;  
            }  
        }  
    }  
    for(i=0;i<n/2;i++)  
        printf("%d ",a[i]);
```

```
for(i=n-1;i>=n/2;i--)  
    printf("%d ",a[i]);  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n=obj.nextInt();
```

```
        int a[]=new int[n];
```

```
        int i;
```

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

```
            a[i]=obj.nextInt();
```

```
        Arrays.sort(a);
```

```
        for(i=0;i<n/2;i++)
```

```
            System.out.print(a[i]+" ");
```

```
for(i=n-1;i>=n/2;i--)  
    System.out.print(a[i]+" ");  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
for i in range(0,n//2):  
    print(L[i],end=' ')  
for i in range(n-1,(n//2)-1,-1):  
    print(L[i],end=' ')
```

LBP187

last and second-last

Given a string, create a new string made up of its last two letters, reversed and seperated by comma.

input -----> a string from the user

con -----> no

output -----> comma seperated last and second-last character

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int n;
```

```
    scanf("%s",s);
```

```
    n=strlen(s);
```

```
    printf("%c,%c",s[n-1],s[n-2]);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s = obj.nextLine();
```

```
        int n = s.length();
```

```
        System.out.println(s.charAt(n-1)+","+s.charAt(n-2));
```

```
    }
```

```
}
```

python implementation:

```
s=input()
```

```
print(s[-1],s[-2],sep=',')
```

LBP188

digital root

Write a program to find the digital root of a given number.

Digital root is the recursive sum of digits in the given number (until single digit is arrived)

input ----> a number from the user

con -----> no

output ---> single digit number

$$129 = 1+2+9 = 12 = 1+2 = 3$$

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int sum(int n)
{
    int sum=0;

    int d;

    while(n!=0)
    {
        d=n%10;
```

```
sum=sum+d;

n=n/10;

}

return sum;

}

int main() {

    int n;

    scanf("%d",&n);

    while(1)

    {

        if(n>=0 && n<=9)

        {

            printf("%d",n);

            break;

        }

        else

            n=sum(n);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    static int sum(int n)

    {

        int sum=0;

        int d;

        while(n!=0)

        {

            d=n%10;

            sum=sum+d;

            n=n/10;

        }

        return sum;

    }

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        while(true){

            if(n>=0 && n<=9)

            {

                System.out.println(n);
```

```
        break;
    }
    else
        n=sum(n);
    }
}
}
```

python implementation:

```
def sum(n):
    sum=0
    while n!=0:
        d=n%10
        sum=sum+d
        n=n//10
    return sum
```

```
n=int(input())
while True:
    if n>=0 and n<=9:
        print(n)
        break
```

else:

n=sum(n)

LBP189

absolute difference

Write a program to find the absolute difference between the original number and its reserved number.

input -----> a number from the user

con -----> no

output ----> absolute difference

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int rev(int n)
```

```
{
```

```
    int r=0;
```

```
int d;

while(n!=0)

{

    d=n%10;

    r=r*10+d;

    n=n/10;

}

return r;

}

int main() {

    int n;

    scanf("%d",&n);

    printf("%d",abs(n-rev(n)));

    return 0;

}
```

java implementation:

```
-----

import java.io.*;

import java.util.*;

public class Solution {

    static int rev(int n)
```



```
{  
    int r=0;  
    int d;  
    while(n!=0)  
    {  
        d=n%10;  
        r=r*10+d;  
        n=n/10;  
    }  
    return r;  
}  
  
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt();  
    System.out.println(Math.abs(n-rev(n)));  
}  
}
```

python implementation:

```
-----  
  
n=input()  
  
print(abs(int(n)-int(n[::-1])))
```

LBP190

lucky draw

A person went to an exhibition.

A lucky draw is going on, where one should buy a ticket.

And if their ticket number appears on the screen, that ticket will be considered as jackpot winner.

He knows the secret of picking up the ticket number, which will be considered for the jackpot.

1. Sort the ticket number in the increasing order.
2. Now, the difference between the adjacent digits should not be more than 2.

If his ticket follows the above condition, then there are more chances to win the jackpot.

Given a ticket number, find whether the ticket is eligible to be part of jackpot or not. Print "Yes/No" based on the result.

input -----> ticket number

con -----> no

output ----> Yes or No

171 ---> 117 ---> 0,6 ----> No

123 ---> 123 ---> 1,1 ----> Yes

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,a[100],i,j,t,flag,len;

    scanf("%d",&n);

    i=0;

    while(n!=0)

    {

        a[i++]=n%10;

        n=n/10;

    }

    len=i;

    for(i=0;i<len;i++)

    {
```

```
for(j=i+1;j<len;j++)
{
    if(a[i]>a[j])
    {
        t=a[i];
        a[i]=a[j];
        a[j]=t;
    }
}

flag=1;
for(i=0;i<len-1;i++)
{
    if(a[i+1]-a[i]>2)
    {
        flag=0;
        break;
    }
}

printf((flag==1)?"Yes":"No");

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int n,len,i=0;
```

```
        n=obj.nextInt();
```

```
        int a[]=new int[3];
```

```
        while(n!=0)
```

```
        {
```

```
            a[i++]=n%10;
```

```
            n=n/10;
```

```
        }
```

```
        len=a.length;
```

```
        Arrays.sort(a);
```

```
        boolean flag=true;
```

```
        for(i=0;i<len-1;i++)
```

```
        {
```

```
        if(a[i+1]-a[i]>2)
        {
            flag=false;
            break;
        }
    }
    System.out.println(flag?"Yes":"No");
}
```

python implementation:

```
L=[int(i) for i in input()]
L.sort()
flag=True
for i in range(len(L)-1):
    if L[i+1]-L[i]>2:
        flag=False
        break
print("Yes" if flag==True else "No")
```

LBP191

test paper set

In an online exam, the test paper set is categorized by the letters A-Z. The students enrolled in the exam have been assigned a numeric value called application ID. To assign the test set to the student, firstly the sum of all digits in the application ID is calculated. If the sum is within the numeric range 1-26 the corresponding alphabetic set code is assigned to the student, else the sum of the digits are calculated again and so on until the sum falls within the 1-26 range.

input -----> application id as int

con -----> no

output -----> set number

A-1,B-2,C-3,D-4,E-5,F-6

123=1+2+3=6 -----> F

>=1 and <=26 else continue

c implementation:

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

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

```
#include <math.h>

#include <stdlib.h>

int sum(int n)
{
    int sum=0;
    int d;
    while(n!=0)
    {
        d=n%10;
        sum=sum+d;
        n=n/10;
    }
    return sum;
}

int main() {
    int n;
    scanf("%d",&n);
    while(1)
    {
        if(n>=1 && n<=26)
        {
            printf("%c",64+n);
            break;
        }
    }
}
```



```
}  
  
else  
  
    n=sum(n);  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    static int sum(int n)  
  
    {  
  
        int sum=0;  
  
        int d;  
  
        while(n!=0)  
  
        {  
  
            d=n%10;  
  
            sum=sum+d;  
  
            n=n/10;  
  
        }
```

```
        return sum;
    }

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        while(true){

            if(n>=1 && n<=26)

            {

                System.out.println((char)(n+64));

                break;

            }

            else

                n=sum(n);

        }

    }
}
```

python implementation:

```
def sum(n):

    sum=0

    while n!=0:

        d=n%10
```

```
sum=sum+d
```

```
n=n//10
```

```
return sum
```

```
n=int(input())
```

```
while True:
```

```
    if n>=1 and n<=26:
```

```
        print(chr(n+64))
```

```
        break
```

```
    else:
```

```
        n=sum(n)
```

LBP192

digits raised to the third power

Cristina appeared for a corporate Hackathon. She cleared first round of this and would like to take next challenge which is coding round. The problem statement comes to her is

"Write a program to find numbers which are in between integer 2 and integer N and such that the sum of its digits raised to the third power is equal to the number with the input given.

input ----> integer N

con -----> no

output ---> an integer value

amstrong number

$123 = 1^3 + 2^3 + 3^3 = 1+8+27= 36$ No

$153 = 1^3 + 5^3 + 3^3 = 1+125+27=153$ Yes

1-----N

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int sum(int n)
```

```
{
```

```
    int s=0;
```

```
    int d;
```

```
    while(n!=0)
```

```
{
```

```
d=n%10;

s=s+(d*d*d);

n=n/10;

}

return s;

}

int main() {

    int n,i;

    scanf("%d",&n);

    for(i=2;i<=n;i++)

    {

        if(i==sum(i))

            printf("%d ",i);

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
    static int sum(int n)  
    {  
        int sum=0;  
        int d;  
        while(n!=0)  
        {  
            d=n%10;  
            sum=sum+(d*d*d);  
            n=n/10;  
        }  
        return sum;  
    }  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n = obj.nextInt(),i;  
        for(i=2;i<=n;i++)  
        {  
            if(i==sum(i))  
                System.out.print(i+" ");  
        }  
    }  
}
```

python implementation:

```
def sum(n):
```

```
    s=0
```

```
    while n!=0:
```

```
        d=n%10
```

```
        s=s+(d*d*d)
```

```
        n=n//10
```

```
    return s
```

```
n=int(input())
```

```
for i in range(2,n+1):
```

```
    if i==sum(i):
```

```
        print(i,end=' ')
```

LBP193

Grocery Shop

There was a grocery shop. Shopkeeper would like to keep transactions as simple as he can. Hence, he used to take money as whole number. To optimize transactions, he decided if someone buys groceries from his shop, he will round money to the nearest whole number having zeros as last digit. Write a program to help shopkeeper to make transactions much simple.

input ----> a number

con -----> no

output ----> nearest int value which ending with 0

45 ----> 50

51 ----> 60

80 ----> 80

49 ----> 50

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```



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

while(1)

{

    if(n%10==0)

    {

        printf("%d",n);

        break;

    }

    else

        n++;

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
int n=obj.nextInt();  
while(true)  
{  
    if(n%10==0)  
    {  
        System.out.println(n);  
        break;  
    }  
    else  
        n++;  
}  
}
```

python implementation:

```
n=int(input())  
while True:  
    if n%10==0:  
        print(n)  
        break  
    else:  
        n=n+1
```

LBP194

Password change

Prakash a technical person wants to update his password for every 15 days, to create a new password, he is converting all lower case letters to upper case and upper case letters to lower case, help prakash to update password.

input -----> a string from the user old password

con -----> no

output -----> updated password

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    scanf("%s",s);
```

```
    for(int i=0;s[i];i++)
```

```
{  
    if(s[i]>='a'&& s[i]<='z')  
        printf("%c",s[i]-32);  
    else if(s[i]>='A'&& s[i]<='Z')  
        printf("%c",s[i]+32);  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        for(int i=0;i<s.length();i++)  
        {  
            char ch = s.charAt(i);  
            if(ch>='a'&& ch<='z')
```

```
System.out.print((char)(ch-32));  
  
else if(ch>='A'&&ch<='Z')  
  
System.out.print((char)(ch+32));  
  
}  
  
}  
  
}
```

python implementation:

```
-----  
  
s=input()  
  
print(s.swapcase())
```

LBP195

Video share

Video share is an online video sharing platform. The company has decided to rate its users channels based on the sum total of the number of views received online and the subscribers. This sum total is referred to as user points. The rating will be given according to the below charts.

User points rating

30-50 Average

51-60 Good

61-80 Excellent

81-100 Outstanding

input -----> points value

con -----> points>=30 and points<=100

output -----> rating

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

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

```
    if(n>=30 && n<=100){
```

```
        if(n>=30 && n<=50)
```

```
            printf("Average");
```

```
        else if(n>=51 && n<=60)
```

```
            printf("Good");
```

```
        else if(n>=61 && n<=80)
```

```
printf("Excellent");  
  
else  
  
printf("Outstanding");  
  
}  
  
else  
  
printf("invalid");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        if(n>=30 && n<=100){  
  
            if(n>=30 && n<=50)  
  
                System.out.println("Average");  
  
            else if(n>=51 && n<=60)
```

```
        System.out.println("Good");

    else if(n>=61 && n<=80)

        System.out.println("Excellent");

    else

        System.out.println("Outstanding");

    }

    else

        System.out.println("invalid");

    }

}
```

python implementation:

```
n=int(input())

if n>=30 and n<=100:

    if n>=30 and n<=50:

        print("Average")

    elif n>=51 and n<=60:

        print("Good")

    elif n>=61 and n<=80:

        print("Excellent")

    else:
```



```
print("Outstanding")  
  
else:  
  
    print("invalid")
```

LBP196

modular exponentiation

Given three numbers b,e, and m.

fill in a function that takes these three positive integer values and outputs $b^e \text{ mod } m$.

input -----> b,e and m values

con -----> no

output -----> $b^e \text{ mod } m$

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int b,e,m;  
    scanf("%d %d %d",&b,&e,&m);  
    printf("%d",(int)pow(b,e)%m);  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int b=obj.nextInt();
```

```
        int e=obj.nextInt();
```

```
        int m=obj.nextInt();
```

```
System.out.println((int)Math.pow(b,e)%m);  
  
}  
  
}
```

python implementation:

```
b,e,m=(int(i) for i in input().split())  
  
print(b**e%m)
```

LBP197

Backspace String Compare

Two strings are said to be the same if they are of the same length and have the same character at each index. Backspacing in a string removes the previous character in the string.

Given two strings containing lowercase english letters and the character '#' which represents a backspace key. determine if the two final strings are equal or not. Return 1 if they are equal else 0.

input ----> two strings s1 and s2

con -----> no

output ----> 1 or 0

ab#c ----> ac

acr# ----> ac

1

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s1[100],s2[100],s3[100]="\0",s4[100]="\0";
```

```
    int i,j;
```

```
    scanf("%s",s1);
```

```
    scanf("%s",s2);
```

```
    j=0;
```

```
    for(i=0;s1[i];i++)
```

```
    {
```

```
        if(s1[i]!='#' && s1[i+1]!='#')
```

```
            s3[j++]=s1[i];
```

```
}

j=0;
for(i=0;s2[i];i++)
{
    if(s2[i]!='#' && s2[i+1]!='#')
        s4[j++]=s2[i];
}

printf((strcmp(s3,s4)==0)?"1":"0");

return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        String s1 = obj.nextLine();
```

```
        String s2 = obj.nextLine();
```

```
StringBuffer sb1 = new StringBuffer();

StringBuffer sb2 = new StringBuffer();

for(int i=0;i<s1.length()-1;i++)
{
    if(s1.charAt(i)!='#' && s1.charAt(i+1)!='#')
        sb1.append(s1.charAt(i));
}

sb1.append(s1.charAt(s1.length()-1));

for(int i=0;i<s2.length()-1;i++)
{
    if(s2.charAt(i)!='#' && s2.charAt(i+1)!='#')
        sb2.append(s2.charAt(i));
}

sb2.append(s2.charAt(s2.length()-1));

System.out.println(sb1.toString().equals(sb2.toString())?"1":"0");

}
```

python implementation:

```
s1=input()
```

```
s2=input()
```

```
s3=[]
```

```
s4=[]

for i in range(len(s1)-1):
    if s1[i]!='#' and s1[i+1]!='#':
        s3.append(s1[i])

for i in range(len(s2)-1):
    if s2[i]!='#' and s2[i+1]!='#':
        s4.append(s2[i])

print("1" if s3==s4 else "0")
```

LBP198

token number

Write an algorithm to generate the token number from the application ID by doing this modifications.

R1. If the digit is even add 1 to it.

R2. If the digit is odd sub 1 from it.

input -----> a number from the user

con-----> no

output ----> token number

45789==> 54698

98754==>

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int rev(int n)
```

```
{
```

```
    int d;
```

```
    int r=0;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        r=r*10+d;
```



```
n=n/10;

}

return r;

}

int main() {

    int n,d;

    scanf("%d",&n);

    n=rev(n);

    while(n!=0)

    {

        d=n%10;

        if(d%2==0)

            printf("%d",d+1);

        else

            printf("%d",d-1);

        n=n/10;

    }

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n = Integer.parseInt(new
StringBuffer(obj.nextLine()).reverse().toString());

        int d;

        while(n!=0)

        {

            d=n%10;

            if(d%2==0)

                System.out.print(d+1);

            else

                System.out.print(d-1);

            n=n/10;

        }

    }

}
```

python implementation:

```
-----  
n=int(input()[::-1])  
while n!=0:  
    d=n%10  
    if d%2==0:  
        print(d+1,end="")  
    else:  
        print(d-1,end="")  
    n=n//10
```

LBP199

score of the player

a game developing company has developed a math game for kids called "Brain Fun". The game is for smartphone users and the player is given list of N positive numbers and a random number K. the player need to divide all the numbers in the list with random number k and then need to add all the quotients received in each division. the sum of all the quotients is the score of the player.

Write an algorithm to generate the score of the player.

input -----> array size, elements and random number k

con -----> no

output ----> an int value

c implementation:

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

```
int main() {
    int n,a[100],k,i,sum=0;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    scanf("%d",&k);
    for(i=0;i<n;i++)
    {
        sum=sum+a[i]/k;
    }
    printf("%d",sum);
    return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,n=obj.nextInt();
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        int k=obj.nextInt();
```

```
        int s=0;
```

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

```
        {
```

```
            s=s+a[i]/k;
```

```
        }
```

```
        System.out.println(s);
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
k=int(input())
```

```
s=0
```

```
for i in L:
```

```
    s=s+i//k
```

```
print(s)
```

LBP200

Perfect Math

Perfect math is an online math program. In once of the assignments the system displays a list of N number and a value K, and students need to calculate the sum of remainders after dividing all the numbers from the list of N numbers by K. The system need to develop a program to calcualte the correct answer for the assignment.

Write an algorithm to calcualte the correct answer for the assignment.

input -----> size N and N elements and K value

con -----> no

output ----> an int value.

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

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

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

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

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

```
    scanf("%d",&k);
```

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

```
    {
```

```
        sum=sum+a[i]%k;
```

```
    }
```

```
    printf("%d",sum);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,n=obj.nextInt();
```

```
        int a[]=new int[n];
```

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

```
            a[i]=obj.nextInt();
```

```
        int k=obj.nextInt();
```

```
        int s=0;
```

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

```
        {
```

```
            s=s+a[i]%k;
```

```
        }
```

```
        System.out.println(s);
```

```
}
```



```
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
k=int(input())
```

```
s=0
```

```
for i in L:
```

```
    s=s+i%k
```

```
print(s)
```

<https://www.hackerrank.com/matrixtest>

1 2 3

4 5 6

7 8 9

order of matrix: 3x3

3 rows

3 cols

1 2 3 4

5 6 7 8

order of matrix: 2x4

10

order of matrix: 1x1

```
int a[10];
```

```
int a[10][10];
```

```
int a[][] = new int[5][5];
```

List only ==> nested list

```
[[1,2,3],[4,5,6],[7,8,9]]
```

1 2 3

4 5 6

7 8 9

1 2 3

4

7 8

array of arrays

nested list

```
int a[] = new int[rows];
```

```
int a[] = new int[3];
```

```
a[0] = new int[3];
```

```
a[1] = new int[1];
```

```
a[2] = new int[2];
```

```
L=[[x,x,x],[x],[x,x]]
```

Hear of this entire matrix is rows x columns

very very imp elements are 1) rows ---> n ---> 3

2) cols ---> m ---> 3

```
int a[5];
```

```
for(i=0;i<5;i++)
```

```
{
```

```
    a[i]
```

```
}
```

```
int a[5][5];
```

```
for(i=0;i<5;i++)
```

```
{
```

```
    for(j=0;j<5;j++)
```

```
    {
```

```
        a[i][j]
```

```
    }
```

```
}
```

Ex:

```
int a[3][2];
```

```
for(i=0;i<3;i++) //3 times
```

```
{
```

```
for(j=0;j<2;j++) //2 times
{
    a[i][j]
}
}
```

i=0 =====> j=0,j=1

i=1 =====> j=0,j=1

i=2 =====> j=0,j=1

(0,0) (0,1)

(1,0) (1,1)

(2,0) (2,1)

LBP201

read and write matrix elements

Write a program to read matrix and display on the console.

input -----> a 3x3 matrix

constraints --> no

output -----> a 3x3 matrix

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[5][5],i,j,n,m;

    scanf("%d",&n);

    scanf("%d",&m);

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

    {

        for(j=0;j<m;j++)

        {

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

        }

    }

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

    {

        for(j=0;j<m;j++)

        {
```

```
        printf("%d ",a[i][j]);  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();  
        int m=obj.nextInt();  
        int a[][] = new int[n][m];  
        int i,j;  
        for(i=0;i<n;i++)  
        {  
            for(j=0;j<m;j++)
```

```
{  
    a[i][j]=obj.nextInt();  
}  
}  
for(i=0;i<n;i++)  
{  
    for(j=0;j<m;j++)  
    {  
        System.out.print(a[i][j]+" ");  
    }  
    System.out.println();  
}  
}
```

python implementation:

```
n=int(input())  
m=int(input())  
a=[]  
for i in range(n):  
    a.append([int(i) for i in input().split()])
```



```
for i in range(n):  
    for j in range(m):  
        print(a[i][j],end=' ')  
    print()
```

```
[[1,2,3],  
 [4,5,6],  
 [7,8,9]]
```

```
1 2 3 ==> '1 2 3' ==> '1', '2', '3' ==> [1,2,3]
```

```
4 5 6
```

```
7 8 9
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

```
7
```

```
8
```

```
9
```

for i in range(n):

```
a.append([int(i) for i in input().split()])
```

for i in range(n):

```
b=[]
```

for j in range(m):

```
b.append(int(input()))
```

```
a.append(b)
```

"1 2 3".split() ==> 1,2,3

"hi hello how are you".split() ==> hi,hello,how,are,you

LBP202

sum of all matrix elements

Write a program to find sum of all elements in the matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of all elements

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[5][5],i,j,n,m,s;
```

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

```
    scanf("%d",&m);
```

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

```
    {
```

```
        for(j=0;j<m;j++)
```

```
        {
```

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

```
        }
```

```
    }
```

```
    s=0;
```

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

```
    {
```

```
for(j=0;j<m;j++)  
{  
    s=s+a[i][j];  
}  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int m=obj.nextInt();  
  
        int a[][] = new int[n][m];  
  
        int i,j,s;  
  
        for(i=0;i<n;i++)
```

```
{  
    for(j=0;j<m;j++)  
    {  
        a[i][j]=obj.nextInt();  
    }  
}  
  
s=0;  
  
for(i=0;i<n;i++)  
{  
    for(j=0;j<m;j++)  
    {  
        s=s+a[i][j];  
    }  
}  
  
System.out.println(s);  
}  
}
```

python implementation:

```
n=int(input())
```

```
m=int(input())
```

```
a=[]
```

```
for i in range(n):  
    a.append([int(i) for i in input().split()])  
  
s=0  
  
for i in range(n):  
    for j in range(m):  
        s=s+a[i][j]  
  
print(s)
```

LBP203

sum of all even elements

Write a program to find sum of all even elements in the matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of all even elements

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],i,j,s;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    s=0;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(a[i][j]%2==0)  
                s=s+a[i][j];  
        }  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int a[][] = new int[3][3];
```

```
        int i,j,s;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        s=0;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```



```
{  
    if(a[i][j]%2==0)  
        s=s+a[i][j];  
}  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
-----  
a=[]  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
s=0  
for i in range(3):  
    for j in range(3):  
        if a[i][j]%2==0:  
            s=s+a[i][j]  
print(s)
```

LBP204

sum of all odd elements

Write a program to find sum of all odd elements in the matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of all odd elements

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

```
}  
  
}  
  
s=0;  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(a[i][j]%2!=0)  
            s=s+a[i][j];  
    }  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

int a[][] = new int[3][3];

int i,j,s;

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        a[i][j]=obj.nextInt();
    }
}

s=0;

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(a[i][j]%2!=0)
            s=s+a[i][j];
    }
}

System.out.println(s);
}
```

python implementation:

```
-----  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
s=0  
  
for i in range(3):  
    for j in range(3):  
        if a[i][j]%2!=0:  
            s=s+a[i][j]  
  
print(s)
```

LBP205

sum of all prime elements

Write a program to find sum of all prime elements in the matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of all prime elements

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int isprime(int n)
{
    int i,f=0;
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
            f++;
    }
    return f==2;
}

int main() {
    int a[3][3],i,j,s;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
```

```
}  
  
}  
  
s=0;  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(isprime(a[i][j]))  
            s=s+a[i][j];  
    }  
}  
printf("%d",s);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {  
    static boolean isprime(int n)  
    {
```

```
        int i,f=0;

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

        {

            if(n%i==0)

                f++;

        }

        return f==2;

    }

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int a[][] = new int[3][3];

    int i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            a[i][j]=obj.nextInt();

        }

    }

    s=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)
```



```
{  
    if(isprime(a[i][j]))  
        s=s+a[i][j];  
}  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
def isprime(n):
```

```
    f=0
```

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

```
        if n%i==0:
```

```
            f=f+1
```

```
    return f==2
```

```
a=[]
```

```
for i in range(3):
```

```
    a.append([int(i) for i in input().split()])
```

```
s=0
```

```
for i in range(3):
```

```
for j in range(3):  
    if isprime(a[i][j]):  
        s=s+a[i][j]  
  
print(s)
```

LBP206

row wise sum in matrix

Write a program to find row wise sum in the matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of each row

c implementation:

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

```
int main() {
```

```
int a[3][3],i,j,s;
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&a[i][j]);
    }
}
for(i=0;i<3;i++)
{
    s=0;
    for(j=0;j<3;j++)
    {
        s=s+a[i][j];
    }
    printf("%d\n",s);
}
return 0;
}
```

java implementation:

import java.io.*;

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][] = new int[3][3];

        int i,j,s;

        for(i=0;i<3;i++)
        {
            for(j=0;j<3;j++)
            {
                a[i][j]=obj.nextInt();
            }
        }

        for(i=0;i<3;i++)
        {
            s=0;
            for(j=0;j<3;j++)
            {
                s=s+a[i][j];
            }

            System.out.println(s);
        }
    }
}
```

```
}  
  
}  
  
}
```

python implementation:

```
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
for i in range(3):  
    s=0  
    for j in range(3):  
        s=s+a[i][j]  
    print(s)
```

LBP207

column wise sum in matrix

Write a program to find column wise sum in the matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of each column

c implementation:

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

```
int main() {
    int a[3][3],i,j,s;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    for(i=0;i<3;i++)
    {
        s=0;
```

```
for(j=0;j<3;j++)  
{  
    s=s+a[j][i];  
}  
printf("%d\n",s);  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int a[][] = new int[3][3];  
        int i,j,s;  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)
```

```
{  
    a[i][j]=obj.nextInt();  
}  
}  
for(i=0;i<3;i++)  
{  
    s=0;  
    for(j=0;j<3;j++)  
    {  
        s=s+a[j][i];  
    }  
    System.out.println(s);  
}  
}
```

python implementation:

```
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
for i in range(3):
```


s=0

for j in range(3):

s=s+a[j][i]

print(s)

LBP208

sum of diagonal elements in matrix

Write a program to find sum of diagonal elements in matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of diagonal elements

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],i,j,s;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    s=0;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(i==j)  
                s=s+a[i][j];  
        }  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int a[][] = new int[3][3];  
  
        int i,j,s;  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
  
        s=0;  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                if(i==j)
```

```
s=s+a[i][j];  
  
    }  
  
}  
  
System.out.println(s);  
  
}  
  
}
```

python implementation:

```
-----  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
s=0  
  
for i in range(3):  
    for j in range(3):  
        if i==j:  
            s=s+a[i][j]  
  
print(s)
```

LBP209

sum of opposite diagonal elements in matrix

Write a program to find sum of opposite diagonal elements in matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of opposite diagonal elements

c implementation:

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

```
int main() {
    int a[3][3],i,j,s;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}
```

```
s=0;

for(i=0;i<3;i++)

{

    s=s+a[i][3-i-1];

}

printf("%d",s);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][] = new int[3][3];

        int i,j,s;

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)
```

```
{  
    a[i][j]=obj.nextInt();  
}  
}  
  
s=0;  
for(i=0;i<3;i++)  
{  
    s=s+a[i][3-i-1];  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
-----  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
s=0  
  
for i in range(3):  
    s=s+a[i][3-i-1]  
  
print(s)
```

LBP210

sum of first and last element in the matrix

Write a program to find sum of first and last element in a matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> sum of first and last element in matrix

c implementation:

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

```
int main() {
    int a[3][3],i,j,s;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
```



```
scanf("%d",&a[i][j]);  
  
}  
  
}  
  
printf("%d",a[0][0]+a[2][2]);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int a[][] = new int[3][3];  
  
        int i,j,s;  
  
        for(i=0;i<3;i++)  
  
        {  
  
            for(j=0;j<3;j++)  
  
            {  
  
                a[i][j]=obj.nextInt();  
  

```

```
}  
  
}  
  
System.out.println(a[0][0]+a[2][2]);  
  
}  
  
}
```

python implementation:

```
-----  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
s=0  
  
print(a[0][0]+a[2][2])
```

LBP211

find the product of diagonal matrix

Write a program to find the product of diagonal matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> find the product of diagonal matrix

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    s=1;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {
```

```
        if(i==j)

            s=s*a[i][j];

        }

    }

    printf("%d",s);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][] = new int[3][3];

        int i,j,s;

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {
```

```
        a[i][j]=obj.nextInt();
    }
}
s=1;
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(i==j)
            s=s*a[i][j];
    }
}
System.out.println(s);
}
```

python implementation:

```
-----
a=[]
for i in range(3):
    a.append([int(i) for i in input().split()])
s=1
for i in range(3):
```

```
for j in range(3):  
    if i==j:  
        s=s*a[i][j]  
  
print(s)
```

LBP212

find the product of opposite diagonal matrix

Write a program to find the product of opposite diagonal matrix.

input -----> a 3x3 matrix

constraints --> no

output -----> find the product of opposite diagonal matrix

c implementation:

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

```
int main() {  
    int a[3][3],i,j,s;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    s=1;  
    for(i=0;i<3;i++)  
    {  
        s=s*a[i][3-i-1];  
    }  
    printf("%d",s);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int a[][] = new int[3][3];  
        int i,j,s;  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
        s=1;  
        for(i=0;i<3;i++)  
        {  
            s=s*a[i][3-i-1];  
        }  
        System.out.println(s);  
    }  
}
```

python implementation:


```
-----  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
s=1  
  
for i in range(3):  
    s=s*a[i][3-i-1]  
  
print(s)
```

LBP213

max element in matrix

Implement a program to print max element in an matrix

input -----> a 3x3 matrix

constraint-> no

output ----> max element in matrix

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>


int main() {

    int a[3][3],i,j,max;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    max=a[0][0];

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if(max<a[i][j])

                max=a[i][j];

        }

    }

}
```

```
printf("%d",max);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,j,max;
```

```
        int a[][]=new int[3][3];
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        max=a[0][0];
```

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(max<a[i][j])  
            max=a[i][j];  
    }  
}  
System.out.println(max);  
}  
}
```

python implementation:

```
-----  
a=[]  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
max=a[0][0]  
for i in range(3):  
    for j in range(3):  
        if max<a[i][j]:  
            max=a[i][j]  
print(max)
```

LBP214

min element in matrix

Implement a program to print min element in an matrix

input ----> a 3x3 matrix

constraint-> no

output ----> min element in matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,min;
```

```
    for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        scanf("%d",&a[i][j]);  
    }  
}  
min=a[0][0];  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(min>a[i][j])  
            min=a[i][j];  
    }  
}  
printf("%d",min);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int i,j,min;  
  
        int a[][]=new int[3][3];  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
  
        min=a[0][0];  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                if(min>a[i][j])  
                    min=a[i][j];  
            }  
        }  
    }  
}
```

```
        System.out.println(min);  
    }  
}
```

python implementation:

```
-----  
  
a=[]  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
min=a[0][0]  
for i in range(3):  
    for j in range(3):  
        if min>a[i][j]:  
            min=a[i][j]  
  
print(min)
```

LBP215

max element in each row of a matrix

Implement a program to print max element in each row of a matrix

input -----> a 3x3 matrix

constraint-> no

output ----> print max element in each row of a matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,max;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

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

```
        }
```

```
    }
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
max=a[i][0];  
for(j=0;j<3;j++)  
{  
    if(max<a[i][j])  
        max=a[i][j];  
}  
printf("%d\n",max);  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,j,max;
```

```
        int a[][]=new int[3][3];
```

```
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        a[i][j]=obj.nextInt();  
    }  
}  
for(i=0;i<3;i++)  
{  
    max=a[i][0];  
    for(j=0;j<3;j++)  
    {  
        if(max<a[i][j])  
            max=a[i][j];  
    }  
    System.out.println(max);  
}  
}
```

python implementation:

a=[]

for i in range(3):

```
a.append([int(i) for i in input().split()])
```

```
for i in range(3):
```

```
    max=a[i][0]
```

```
    for j in range(3):
```

```
        if max<a[i][j]:
```

```
            max=a[i][j]
```

```
print(max)
```

LBP216

min element in each row of a matrix

Implement a program to print min element in each row of a matrix

input -----> a 3x3 matrix

constraint-> no

output ----> print min element in each row of a matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,min;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

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

```
        }
```

```
    }
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        min=a[i][0];
```

```
        for(j=0;j<3;j++)
```

```
        {
```

```
            if(min>a[i][j])
```

```
                min=a[i][j];
```

```
        }
```

```
        printf("%d\n",min);
```

```
    }
```

```
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int i,j,min;  
  
        int a[][]=new int[3][3];  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
  
        for(i=0;i<3;i++)  
        {
```

```
min=a[i][0];  
for(j=0;j<3;j++)  
{  
    if(min>a[i][j])  
        min=a[i][j];  
}  
System.out.println(min);  
}  
}
```

python implementation:

```
a=[]  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
for i in range(3):  
    min=a[i][0]  
    for j in range(3):  
        if min>a[i][j]:  
            min=a[i][j]  
    print(min)
```

LBP217

transpose of the given matrix

Implement a program to print transpose of a matrix

input ----> a 3x3 matrix

constraint-> no

output ----> print transpose of the matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j;
```

```
    for(i=0;i<3;i++)
```

```
{
```



```
for(j=0;j<3;j++)  
{  
    scanf("%d",&a[i][j]);  
}  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",a[j][i]);  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
  
    int i,j;  
  
    int a[][]=new int[3][3];  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            System.out.print(a[j][i]+" ");  
        }  
        System.out.println();  
    }  
}
```

python implementation:

```
-----  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
for i in range(3):  
    for j in range(3):  
        print(a[j][i],end=' ')  
  
    print()
```

LBP218

trace of the given matrix

Implement a program to find trace(sum of diagonal elements) of the given matrix.

input -----> a 3x3 matrix

constraint-> no

output ----> print trace of the matrix

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    s=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if(i==j)

                s=s+a[i][j];

        }

    }

}
```

```
printf("%d",s);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int a[][] = new int[3][3];  
  
        int i,j,s;  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
  
        s=0;
```

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(i==j)  
            s=s+a[i][j];  
    }  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
a=[]  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
s=0  
for i in range(3):  
    for j in range(3):  
        if i==j:  
            s=s+a[i][j]  
print(s)
```

LBP219

find the frequency of odd and even

Write a program to find frequency of odd and even elements in the matrix excluding 0.

input -----> a 3x3 matrix

constraint -> no

output -----> odd elements and even elements count

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],i,j,c1,c2;

    for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        scanf("%d",&a[i][j]);  
    }  
}  
  
c1=0;//odd  
c2=0;//even  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(a[i][j]%2==0 && a[i][j]!=0)  
            c2++;  
        if(a[i][j]%2!=0)  
            c1++;  
    }  
}  
  
printf("%d\n%d",c1,c2);  
  
return 0;  
}
```


java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,j,c1,c2;
```

```
        int a[][]=new int[3][3];
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        c1=0;//odd
```

```
        c2=0;//even
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
{  
    if(a[i][j]%2==0 && a[i][j]!=0)  
        c2++;  
    if(a[i][j]%2!=0)  
        c1++;  
}  
}  
System.out.println(c1);  
System.out.println(c2);  
}  
}
```

python implementation:

```
-----  
a=[]  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
c1=0#odd  
c2=0#even  
for i in range(3):  
    for j in range(3):  
        if a[i][j]%2==0 and a[i][j]!=0:  
            c2=c2+1
```

```
if a[i][j]%2!=0:
```

```
    c1=c1+1
```

```
print(c1)
```

```
print(c2)
```

LBP220

identity matrix

Implement a program to check whether the given matrix is identity matrix or not

input -----> a 3x3 matrix

constraint --> no

output -----> Yes or No

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],i,j,flag=1;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(i==j && a[i][j]!=1)  
            {  
                flag=0;  
                break;  
            }  
            if(i!=j && a[i][j]!=0)  
            {  
                flag=0;  
                break;  
            }  
        }  
    }  
}
```

```
}  
  
}  
  
printf((flag==1)?"Yes":"No");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int i,j;  
  
        boolean flag=true;  
  
        int a[][]=new int[3][3];  
  
        for(i=0;i<3;i++)  
  
        {  
  
            for(j=0;j<3;j++)  
  
            {  
  
                a[i][j]=obj.nextInt();
```

```
    }  
  
    }  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(i==j && a[i][j]!=1)  
            {  
                flag=false;  
                break;  
            }  
            if(i!=j && a[i][j]!=0)  
            {  
                flag=false;  
                break;  
            }  
        }  
    }  
  
    System.out.println((flag)?"Yes":"No");  
  
    }  
}
```

python implementation:

```
-----  
flag=True  
  
a=[]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
for i in range(3):  
    for j in range(3):  
        if i==j and a[i][j]!=1:  
            flag=False  
            break  
        if i!=j and a[i][j]!=0:  
            flag=False  
            break  
  
print("Yes" if flag else "No")
```

LBP221

two matrices are equal or not

Implement a program to check whether the given matrices are equal or not

input -----> a 3x3 matrix

constraint --> no

output -----> Yes or No

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],b[3][3],i,j,flag=1;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d",&b[i][j]);
```



```
}  
  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(a[i][j]!=b[i][j])  
        {  
            flag=0;  
            break;  
        }  
    }  
}  
  
printf((flag==1)?"Yes":"No");  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    int i,j;  
  
    boolean flag=true;  
  
    int a[][]=new int[3][3];  
    int b[][]=new int[3][3];  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            b[i][j]=obj.nextInt();  
        }  
    }  
  
    for(i=0;i<3;i++)  
    {
```

```
for(j=0;j<3;j++)
{
    if(a[i][j]!=b[i][j])
    {
        flag=false;
        break;
    }
}

System.out.println((flag?"Yes":"No"));
}
```

python implementation:

```
-----

flag=True

a=[]
b=[]

for i in range(3):
    a.append([int(i) for i in input().split()])

for i in range(3):
    b.append([int(i) for i in input().split()])

for i in range(3):
```

```
for j in range(3):  
    if a[i][j]!=b[i][j]:  
        flag=False  
        break  
print("Yes" if flag else "No")
```

LBP222

addition of two matrices

Write a program to perform addition operation on two matrices

input -----> two 3x3 matrices

constraint----> no

output -----> resultant matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],b[3][3],c[3][3],i,j;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&b[i][j]);  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            c[i][j]=a[i][j]+b[i][j];  
        }  
    }
```

```
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",c[i][j]);  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,j;
```

```
        boolean flag=true;
```

```
int a[][]=new int[3][3];

int b[][]=new int[3][3];

int c[][]=new int[3][3];

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        a[i][j]=obj.nextInt();

    }

}

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        b[i][j]=obj.nextInt();

    }

}

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        c[i][j]=a[i][j]+b[i][j];

    }

}
```

```
}  
  
for(i=0;i<3;i++)  
  
{  
  
    for(j=0;j<3;j++)  
  
    {  
  
        System.out.print(c[i][j]+" ");  
  
    }  
  
    System.out.println();  
  
}  
  
}
```

python implementation:

```
-----  
  
a=[]  
  
b=[]  
  
c=[]  
  
for i in range(3):  
  
    a.append([int(i) for i in input().split()])  
  
for i in range(3):  
  
    b.append([int(i) for i in input().split()])  
  
for i in range(3):
```



```
cc=[]  
  
for j in range(3):  
    cc.append(a[i][j]+b[i][j])  
  
c.append(cc)  
  
for i in range(3):  
    for j in range(3):  
        print(c[i][j],end=' ')  
  
    print()
```

LBP223

subtraction of two matrices

Write a program to perform subtraction operation on two matrices

input -----> two 3x3 matrices

constraint----> no

output -----> resultant matrix

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>


int main() {

    int a[3][3],b[3][3],c[3][3],i,j;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    for(i=0;i<3;i++)

    {
```

```
    for(j=0;j<3;j++)  
    {  
        c[i][j]=a[i][j]-b[i][j];  
    }  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",c[i][j]);  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {
```

```
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    int i,j;  
  
    boolean flag=true;  
  
    int a[][]=new int[3][3];  
    int b[][]=new int[3][3];  
    int c[][]=new int[3][3];  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            b[i][j]=obj.nextInt();  
        }  
    }  
  
    for(i=0;i<3;i++)  
    {
```

```
for(j=0;j<3;j++)
{
    c[i][j]=a[i][j]-b[i][j];
}
}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        System.out.print(c[i][j]+" ");
    }
    System.out.println();
}
}
```

python implementation:

a=[]

b=[]

c=[]

for i in range(3):

```
a.append([int(i) for i in input().split()])  
  
for i in range(3):  
  
    b.append([int(i) for i in input().split()])  
  
for i in range(3):  
  
    cc=[]  
  
    for j in range(3):  
  
        cc.append(a[i][j]-b[i][j])  
  
    c.append(cc)  
  
for i in range(3):  
  
    for j in range(3):  
  
        print(c[i][j],end=' ')  
  
    print()
```

LBP224

multiplication of two matrices

Write a program to perform multiplication operation on two matrices

input -----> two 3x3 matrices

constraint----> no

output -----> resultant matrix

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],b[3][3],i,j,c[3][3],k;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

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

        }

    }

}
```

```
}  
  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        c[i][j]=0;  
        for(k=0;k<3;k++)  
        {  
            c[i][j]=c[i][j]+(a[i][k]*b[k][j]);  
        }  
    }  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",c[i][j]);  
    }  
    printf("\n");  
}  
  
return 0;  
  
}
```


java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int i,j,k;
```

```
        int a[][]=new int[3][3];
```

```
        int b[][]=new int[3][3];
```

```
        int c[][]=new int[3][3];
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        b[i][j]=obj.nextInt();  
    }  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        c[i][j]=0;  
        for(k=0;k<3;k++)  
        {  
            c[i][j]=c[i][j]+(a[i][k]*b[k][j]);  
        }  
    }  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print(c[i][j]+" ");  
    }  
}
```

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

python implementation:

```
-----  
  
a=[]  
b=[]  
c=[[0,0,0],[0,0,0],[0,0,0]]  
  
for i in range(3):  
    a.append([int(i) for i in input().split()])  
  
for i in range(3):  
    b.append([int(i) for i in input().split()])  
  
  
for i in range(3):  
    for j in range(3):  
        for k in range(3):  
            c[i][j]=c[i][j]+(a[i][k]*b[k][j])  
  
  
for i in range(3):  
    for j in range(3):
```

```
print(c[i][j],end=' ')
```

```
print()
```

LBP225

sort all the elements in a matrix in asc order

Implement a program to sort all the elements in asc order in the matrix

input -----> a matrix

cons-----> no

output -----> result matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
int a[3][3],i,j,t,k,b[100];

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

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

    }

}

k=0;

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        b[k++]=a[i][j];

    }

}

for(i=0;i<k;i++)

{

    for(j=i+1;j<k;j++)

    {

        if(b[i]>b[j])

        {

            t=b[i];
```

```
        b[i]=b[j];

        b[j]=t;

    }

}

}

k=0;

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        a[i][j]=b[k++];

    }

}

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        printf("%d ",a[i][j]);

    }

    printf("\n");

}

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int a[][]=new int[3][3];
```

```
        int i,j,k;
```

```
        int b[]=new int[9];
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        k=0;
```

```
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        b[k++]=a[i][j];  
    }  
}  
Arrays.sort(b);  
k=0;  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        a[i][j]=b[k++];  
    }  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print(a[i][j]+" ");  
    }  
    System.out.println();  
}
```



```
}
```

```
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
ll=[]
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        ll.append(l[i][j])
```

```
ll.sort()
```

```
k=0
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        l[i][j]=ll[k]
```

```
        k=k+1
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        print(l[i][j],end=' ')
```

```
print()
```

LBP226

sort all the elements in a matrix in desc order

Implement a program to sort all the elements in desc in the matrix

input -----> a matrix

cons-----> no

output -----> result matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,t,k,b[100];
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
{  
    scanf("%d",&a[i][j]);  
}  
}  
k=0;  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        b[k++]=a[i][j];  
    }  
}  
for(i=0;i<k;i++)  
{  
    for(j=i+1;j<k;j++)  
    {  
        if(b[i]<b[j])  
        {  
            t=b[i];  
            b[i]=b[j];  
            b[j]=t;  
        }  
    }  
}
```

```
}  
  
k=0;  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        a[i][j]=b[k++];  
    }  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",a[i][j]);  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][]=new int[3][3];

        int i,j,k;

        int b[]=new int[9];

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {

                a[i][j]=obj.nextInt();

            }

        }

        k=0;

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {

                b[k++]=a[i][j];

            }

        }

    }

}
```

```
    }  
  
    }  
  
    Arrays.sort(b);  
  
    k=9-1;  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=b[k--];  
        }  
    }  
  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            System.out.print(a[i][j]+" ");  
        }  
        System.out.println();  
    }  
}  
  
}
```

python implementation:

```
-----  
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
ll=[]  
for i in range(3):  
    for j in range(3):  
        ll.append(l[i][j])  
ll.sort(reverse=True)  
k=0  
for i in range(3):  
    for j in range(3):  
        l[i][j]=ll[k]  
        k=k+1  
for i in range(3):  
    for j in range(3):  
        print(l[i][j],end=' '  
print()
```

LBP227

sort all the elements in a row in asc order

Implement a program to sort all the rowwise elements in asc order

input -----> a matrix

cons-----> no

output -----> result matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,k,t;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

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

```
        }
```



```
}  
  
//logic begin  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        //a[i][j] where i is fixed  
        //a[0][0],a[0][1],a[0][2]  
        for(k=j+1;k<3;k++)  
        {  
            if(a[i][j]>a[i][k])  
            {  
                t=a[i][j];  
                a[i][j]=a[i][k];  
                a[i][k]=t;  
            }  
        }  
    }  
}  
  
//logic ends  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)
```

```
{  
    printf("%d ",a[i][j]);  
}  
printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

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

```
        int a[][]=new int[3][3];
```

```
        int i,j;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
a[i][j]=obj.nextInt();

    }

}

for(i=0;i<3;i++)

{

    Arrays.sort(a[i]);

}

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        System.out.print(a[i][j]+" ");

    }

    System.out.println();

}

}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
for i in range(3):
```

```
    l[i].sort()
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        print(l[i][j],end=' ')
```

```
    print()
```

LBP228

sort all the elements in a row in desc order

Implement a program to sort all the row wise elements in desc order

input -----> a matrix

cons-----> no

output -----> result matrix

c implementation:

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

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

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

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,k,t;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

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

```
        }
```

```
    }
```

```
    //logic begin
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

```
            //a[i][j] where i is fixed
```

```
            //a[0][0],a[0][1],a[0][2]
```

```
            for(k=j+1;k<3;k++)
```

```
            {
```

```
                if(a[i][j]<a[i][k])
```

```
                {
```

```
                    t=a[i][j];
```

```
        a[i][j]=a[i][k];

        a[i][k]=t;

    }

}

}

}

//logic ends

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        printf("%d ",a[i][j]);

    }

    printf("\n");

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int a[][]=new int[3][3];  
        int i,j;  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
        for(i=0;i<3;i++)  
        {  
            Arrays.sort(a[i]);  
        }  
        for(i=0;i<3;i++)  
        {  
            for(j=3-1;j>=0;j--)  
            {  
                System.out.print(a[i][j]+" ");  
            }  
        }  
    }  
}
```

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

python implementation:

```
-----  
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
for i in range(3):  
    l[i].sort(reverse=True)  
for i in range(3):  
    for j in range(3):  
        print(l[i][j],end=' ')  
    print()
```

LBP229

sort all the elements in a column in asc order

Implement a program to sort all the column values in asc order

input -----> a matrix

cons-----> no

output -----> result matrix

logic:

1. read a matrix from the user
2. copy the content of a matrix to b matrix like $b[i][j]=a[j][i]$
3. apply row wise sorting method
4. print $b[j][i]$ matrix

c implementation:

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

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

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],b[3][3],i,j,k,t;
```

```
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        scanf("%d",&a[i][j]);
    }
}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        b[i][j]=a[j][i];
    }
}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        for(k=j+1;k<3;k++)
        {
            if(b[i][j]>b[i][k])
            {
                t=b[i][j];
```

```
        b[i][j]=b[i][k];

        b[i][k]=t;

    }

}

}

}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d ",b[j][i]);

    }

    printf("\n");

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][]=new int[3][3];

        int b[][]=new int[3][3];

        int i,j,k;

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {

                a[i][j]=obj.nextInt();

            }

        }

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {

                b[i][j]=a[j][i];

            }

        }

        for(i=0;i<3;i++)

        {
```

```
Arrays.sort(b[i]);  
  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print(b[j][i]+" ");  
    }  
    System.out.println();  
}  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
ll=[[0,0,0],[0,0,0],[0,0,0]]  
for i in range(3):  
    for j in range(3):  
        ll[i][j]=l[j][i]
```

for i in range(3):

```
ll[i].sort()
```

for i in range(3):

```
for j in range(3):
```

```
    print(ll[j][i],end=' ')
```

```
print()
```

LBP230

sort all the elements in a column in desc order

Implement a program to sort the all the column values in desc order

input -----> a matrix

cons-----> no

output -----> result matrix

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],b[3][3],i,j,k,t;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            b[i][j]=a[j][i];  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            for(k=j+1;k<3;k++)
```

```
{
    if(b[i][j]<b[i][k])
    {
        t=b[i][j];
        b[i][j]=b[i][k];
        b[i][k]=t;
    }
}

}

}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        printf("%d ",b[j][i]);
    }
    printf("\n");
}

return 0;
}
```

java implementation:


```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int a[][]=new int[3][3];  
        int b[][]=new int[3][3];  
  
        int i,j,k;  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                b[i][j]=a[j][i];  
            }  
        }  
    }  
}
```

```
    }  
  
    }  
  
    for(i=0;i<3;i++)  
    {  
        Arrays.sort(b[i]);  
    }  
  
    for(i=3-1;i>=0;i--)  
    {  
        for(j=0;j<3;j++)  
        {  
            System.out.print(b[j][i]+" ");  
        }  
        System.out.println();  
    }  
    }  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
ll=[[0,0,0],[0,0,0],[0,0,0]]
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        ll[i][j]=l[j][i]
```

```
for i in range(3):
```

```
    ll[i].sort(reverse=True)
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        print(ll[j][i],end=' ')
```

```
    print()
```

LBP231

sparse matrix

Implement a program to check whether the given matrix is sparse matrix or not.

Note: a sparse matrix is a matrix with the majority of its elements equal to zero.

input-----> a matrix

con -----> no

output -----> Yes or No

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],counter,i,j;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

```
            scanf("%d",&a[i][j]);
```

```
        }
```

```
    }
```

```
    counter=0;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
{  
    if(a[i][j]==0)  
        counter++;  
}  
}  
  
printf((counter>=5)?"Yes":"No");  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int i,j,counter=0;
```

```
        int a[][]=new int[3][3];
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
    for(j=0;j<3;j++)
    {
        a[i][j]=obj.nextInt();
    }
}

for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(a[i][j]==0)
            counter++;
    }
}

System.out.println((counter>=5)?"Yes":"No");
}
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
l2=[int(i) for i in input().split()]
l3=[int(i) for i in input().split()]
l=[l1,l2,l3]
```

counter=0

for i in l:

 counter=counter+i.count(0)

print("Yes" if counter>=5 else "No")

LBP232

swaping of two rows

Implement a program to swap two given rows.

input -----> matrix and m and n values

con -----> no

output -----> modified matrix

0 ==> 11 22 33

1 ==> 44 55 66

2 ==> 77 88 99

m=1 and n=2

m-1 and n-1

1-1 and 2-1

0 and 1

0 ==> 44 55 66

1 ==> 11 22 33

2 ==> 77 88 99

for i=0,i<3,i++

t=a[m-1][i]

a[m-1][i]=a[n-1][i]

a[n-1][i]=t

c implementation:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

int a[3][3],i,j,m,n,t;

for(i=0;i<3;i++)

{


```
for(j=0;j<3;j++)  
{  
    scanf("%d",&a[i][j]);  
}  
}  
scanf("%d %d",&m,&n);  
for(i=0;i<3;i++)  
{  
    t=a[m-1][i];  
    a[m-1][i]=a[n-1][i];  
    a[n-1][i]=t;  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",a[i][j]);  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j,t,m,n;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        m=obj.nextInt();
```

```
        n=obj.nextInt();
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
t=a[m-1][i];  
a[m-1][i]=a[n-1][i];  
a[n-1][i]=t;  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print(a[i][j]+" ");  
    }  
    System.out.println();  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
m=int(input())  
n=int(input())  
l=[l1,l2,l3]
```

#swaping

$l[m-1], l[n-1] = l[n-1], l[m-1]$

for i in range(3):

 for j in range(3):

 print(l[i][j], end=' ')

 print()

LBP233

swaping of two columns

Implement a program to swap two given columns

input -----> matrix and m and n values

con -----> no

output -----> modified matrix

c implementation:

#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>

```
int main() {  
    int a[3][3],i,j,m,n,t;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    scanf("%d %d",&m,&n);  
    for(i=0;i<3;i++)  
    {  
        t=a[i][m-1];  
        a[i][m-1]=a[i][n-1];  
        a[i][n-1]=t;  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            printf("%d ",a[i][j]);  
        }  
    }
```

```
printf("\n");  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int a[][]=new int[3][3];  
  
        int i,j,t,m,n;  
  
        for(i=0;i<3;i++)  
  
        {  
  
            for(j=0;j<3;j++)  
  
            {  
  
                a[i][j]=obj.nextInt();  
  
            }  
  
        }  
  
    }
```

```
}  
  
m=obj.nextInt();  
n=obj.nextInt();  
for(i=0;i<3;i++)  
{  
    t=a[i][m-1];  
    a[i][m-1]=a[i][n-1];  
    a[i][n-1]=t;  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print(a[i][j]+" ");  
    }  
    System.out.println();  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
l3=[int(i) for i in input().split()]
m=int(input())
n=int(input())
l=[l1,l2,l3]
#swaping
for i in range(3):
    l[i][m-1],l[i][n-1]=l[i][n-1],l[i][m-1]
for i in range(3):
    for j in range(3):
        print(l[i][j],end=' ')
    print()
```

LBP234

interchange the diagonals

Program to accept a matrix of order 3x3 & interchange the diagonals.

input -----> a 3x3 matrix

con -----> no

output -----> modified matrix

i=0 ==> 1 2 3

i=1 ==> 4 5 6

i=2 ==> 7 8 9

a[i][i]

a[i][3-i-1]

diagonal elements ==> a[0][0],a[1][1],a[2][2] ==> 1, 5, 9

diagonal elements ==> a[0][2],a[1][1],a[2][0] ==> 3, 5, 7

swap a[i][i] and a[i][3-i-1]

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j,t;
```

```
    for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        scanf("%d",&a[i][j]);  
    }  
}  
for(i=0;i<3;i++)  
{  
    t=a[i][i];  
    a[i][i]=a[i][3-i-1];  
    a[i][3-i-1]=t;  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",a[i][j]);  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j,t;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            t=a[i][i];
```

```
            a[i][i]=a[i][3-i-1];
```

```
            a[i][3-i-1]=t;
```

```
}  
  
for(i=0;i<3;i++)  
  
{  
  
    for(j=0;j<3;j++)  
  
    {  
  
        System.out.print(a[i][j]+" ");  
  
    }  
  
    System.out.println();  
  
}  
  
}
```

python implementation:

```
-----  
  
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
  
for i in range(3):  
  
    l[i][i],l[i][3-i-1]=l[i][3-i-1],l[i][i]  
  
for i in range(3):  
  
    for j in range(3):  
  
        print(l[i][j],end=' ')
```

print()

LBP235

upper triangular matrix

Program to accept a matrix and check whether it is upper triangular matrix or not

input -----> a 3x3 matrix

con -----> no

output -----> Yes or No

1 2 3

4 5 6

7 8 9

No

1 2 3

0 5 6

0 0 9

Yes

logic:

(0,0) (0,1) (0,2)

(1,0) (1,1) (1,2)

(2,0) (2,1) (2,2)

flag=1;

(1,0),(2,0),(2,1) ==> j<i && a[i][j]!=0

then set flag=0;

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],i,j,flag=1;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(j<i && a[i][j]!=0)  
            {  
                flag=0;  
            }  
        }  
    }  
    printf((flag==1)?"Yes":"No");  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j;
```

```
        boolean flag=true;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```



```
        if(a[i][j]!=0 && j<i)

            flag=false;

        }

    }

    System.out.println(flag?"Yes":"No");

}

}
```

python implementation:

```
-----

l1=[int(i) for i in input().split()]
l2=[int(i) for i in input().split()]
l3=[int(i) for i in input().split()]
l=[l1,l2,l3]

flag=True

for i in range(3):

    for j in range(3):

        if j<i and l[i][j]!=0:

            flag=False

print("Yes" if flag else "No")
```

LBP236

lower triangular matrix

Program to accept a matrix and check whether it is lower triangular matrix or not

input -----> a 3x3 matrix

con -----> no

output -----> Yes or No

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int a[3][3],i,j,flag=1;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
```

```
scanf("%d",&a[i][j]);  
  
    }  
  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(j>i && a[i][j]!=0)  
        {  
            flag=0;  
        }  
    }  
}  
  
printf((flag==1)?"Yes":"No");  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int a[][]=new int[3][3];  
    int i,j;  
    boolean flag=true;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(a[i][j]!=0 && j>i)  
                flag=false;  
        }  
    }  
    System.out.println(flag?"Yes":"No");  
}
```

}

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
flag=True
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        if j>i and l[i][j]!=0:
```

```
            flag=False
```

```
print("Yes" if flag else "No")
```

LBP237

Scalar matrix multiplication

Implement a program to read a matrix and multiplier and return scalar matrix multiplication.

input -----> a 3x3 matrix and multiplier

con -----> no

output -----> resultant matrix

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int a[3][3],i,j,n;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
    scanf("%d",&n);
    for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",a[i][j]*n);  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j,n;
```

```
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        a[i][j]=obj.nextInt();  
    }  
}  
n=obj.nextInt();  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print((a[i][j]*n)+" ");  
    }  
    System.out.println();  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```



```
l=[l1,l2,l3]
n=int(input())
for i in range(3):
    for j in range(3):
        print(l[i][j]*n,end=' ')
    print()
```

LBP238

Symmetric matrix

Implement a program to read a matrix and check whether the given matrix is symmetric matrix or not

input -----> a 3x3 matrix

con -----> no

output -----> Yes or No

c implementation:

#include <stdio.h>

```
#include <string.h>

#include <math.h>

#include <stdlib.h>

int main() {

    int a[3][3],i,j,c;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    c=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if(a[i][j]==a[j][i])

                c++;

        }

    }

    printf((c==9?"Yes":"No");
```

```
return 0;  
  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j,c=0;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        if(a[i][j]==a[j][i])  
            c++;  
    }  
}  
  
System.out.println((c==9)?"Yes":"No");  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
  
c=0  
  
for i in range(3):  
    for j in range(3):  
        if l[i][j]==l[j][i]:  
            c=c+1  
  
print("Yes" if c==9 else "No")
```

LBP239

print diagonal elements

Implement a program to read a matrix and display only diagonal elements.

input -----> a 3x3 matrix

con -----> no

output -----> print only diagonal elements

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[3][3],i,j;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
{  
    scanf("%d",&a[i][j]);  
}  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(i==j)  
        {  
            printf("%d ",a[i][j]);  
        }  
        else  
            printf(" ");  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

import java.io.*;

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][]=new int[3][3];

        int i,j;

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {

                a[i][j]=obj.nextInt();

            }

        }

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {

                if(i==j)

                    System.out.print(a[i][j]+" ");

                else

                    System.out.print(" ");

            }

        }

    }

}
```

```
}  
  
System.out.println();  
  
}  
  
}  
  
}
```

python implementation:

```
-----  
  
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
  
for i in range(3):  
    for j in range(3):  
        if i==j:  
            print(l[i][j],end=' ')  
        else:  
            print(' ',end='')  
    print()
```

LBP240

Square of Each Element of Matrix

Implement a program to find square of each element present in a matrix.

input -----> a 3x3 matrix

con -----> no

output -----> resultant matrix

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int a[3][3],i,j;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            scanf("%d",&a[i][j]);
        }
    }
}
```

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",a[i][j]*a[i][j]);  
    }  
    printf("\n");  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j;
```

```
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        a[i][j]=obj.nextInt();  
    }  
}  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        System.out.print((a[i][j]*a[i][j])+" ");  
    }  
    System.out.println();  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
for i in range(3):  
    for j in range(3):  
        print(l[i][j]*l[i][j],end=' ')  
    print()
```

LBP241

Sum of even indexed rows in matrix

Implement a program to find sum of even indexed rows in the given matrix.

input -----> a 3x3 matrix

con -----> no

output -----> sum as an int

0 ---> 1 2 3

1----> 4 5 6

2 ---> 7 8 9

$1+2+3+7+8+9=30$

logic

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(i%2==0)  
            s=s+a[i][j];  
    }  
}
```

print s

c implementation:

```
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>
```

```
int main() {  
  
    int a[3][3],i,j,s;  
  
    for(i=0;i<3;i++)  
  
    {  
  
        for(j=0;j<3;j++)
```

```
{  
    scanf("%d",&a[i][j]);  
}  
}  
  
s=0;  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(i%2==0)  
            s=s+a[i][j];  
    }  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
  
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int a[][]=new int[3][3];  
    int i,j,s=0;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(i%2==0)  
                s=s+a[i][j];  
        }  
    }  
    System.out.println(s);  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
s=0
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        if i%2==0:
```

```
            s=s+l[i][j]
```

```
print(s)
```

LBP242

Sum of odd indexed rows in matrix

Implement a program to find sum of odd indexed rows in the given matrix.

input -----> a 3x3 matrix

con -----> no

output -----> sum as an int

0 ---> 1 2 3

1----> 4 5 6

2 ---> 7 8 9

4+5+6 = 15

logic

```
for(i=0;i<3;i++)
```

```
{
```

```
    for(j=0;j<3;j++)
```

```
    {
```

```
        if(i%2!=0)
```

```
            s=s+a[i][j];
```

```
    }
```

```
}
```

print s

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    s=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if(i%2!=0)

                s=s+a[i][j];

        }

    }

}
```

```
printf("%d",s);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int a[][]=new int[3][3];  
  
        int i,j,s=0;  
  
        for(i=0;i<3;i++)  
        {  
            for(j=0;j<3;j++)  
            {  
                a[i][j]=obj.nextInt();  
            }  
        }  
  
        for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        if(i%2!=0)  
            s=s+a[i][j];  
    }  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
s=0  
for i in range(3):  
    for j in range(3):  
        if i%2!=0:  
            s=s+l[i][j]  
print(s)
```

LBP243

Sum of even indexed cols in matrix

Implement a program to find sum of even indexed cols in the given matrix.

input -----> a 3x3 matrix

con -----> no

output -----> sum as an int

0 1 2

0 ---> 1 2 3

1----> 4 5 6

2 ---> 7 8 9

$1+4+7+3+6+9 = 30$

logic

```
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(j%2==0)  
            s=s+a[i][j];  
    }  
}
```

print s

c implementation:

```
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>
```

```
int main() {  
  
    int a[3][3],i,j,s;  
  
    for(i=0;i<3;i++)  
  
    {  
  
        for(j=0;j<3;j++)
```

```
{  
    scanf("%d",&a[i][j]);  
}  
}  
  
s=0;  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(j%2==0)  
            s=s+a[i][j];  
    }  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int a[][]=new int[3][3];  
    int i,j,s=0;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if(j%2==0)  
                s=s+a[i][j];  
        }  
    }  
    System.out.println(s);  
}
```


python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
s=0
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        if j%2==0:
```

```
            s=s+l[i][j]
```

```
print(s)
```

LBP244

Sum of odd indexed cols in matrix

Implement a program to find sum of odd indexed cols in the given matrix.

input -----> a 3x3 matrix

con -----> no

output -----> sum as an int

0 1 2

0 ---> 1 2 3

1----> 4 5 6

2 ---> 7 8 9

2+5+8 =15

logic

```
for(i=0;i<3;i++)
```

```
{
```

```
    for(j=0;j<3;j++)
```

```
    {
```

```
        if(j%2!=0)
```

```
            s=s+a[i][j];
```

```
    }
```

```
}
```

print s

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a[3][3],i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    s=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {
```

```
        if(j%2!=0)

            s=s+a[i][j];

        }

    }

    printf("%d",s);

    return 0;

}
```

java implementation:

```
-----

import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int a[][]=new int[3][3];

        int i,j,s=0;

        for(i=0;i<3;i++)

        {

            for(j=0;j<3;j++)

            {
```

```
        a[i][j]=obj.nextInt();
    }
}
for(i=0;i<3;i++)
{
    for(j=0;j<3;j++)
    {
        if(j%2!=0)
            s=s+a[i][j];
    }
}
System.out.println(s);
}
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
l2=[int(i) for i in input().split()]
l3=[int(i) for i in input().split()]
l=[l1,l2,l3]
s=0
for i in range(3):
```

```
for j in range(3):  
    if j%2!=0:  
        s=s+l[i][j]  
print(s)
```

LBP245

Sum of elements whose sum of row index and col index is even

Implement a program to find sum of row index and col index is even in the given matrix.

input -----> a 3x3 matrix
con -----> no
output -----> sum as an int

formula: $(i+j)\%2==0$

logic

```
for(i=0;i<3;i++)
```

```
{  
    for(j=0;j<3;j++)  
    {  
        if((i+j)%2=0)  
            s=s+a[i][j];  
    }  
}
```

print s

c implementation:

```
#include <stdio.h>  
#include <string.h>  
#include <math.h>  
#include <stdlib.h>
```

```
int main() {  
    int a[3][3],i,j,s;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {
```

```
scanf("%d",&a[i][j]);  
  
    }  
  
}  
  
s=0;  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if((i+j)%2==0)  
            s=s+a[i][j];  
    }  
}  
  
printf("%d",s);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
  
public class Solution {
```



```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int a[][]=new int[3][3];  
    int i,j,s=0;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            if((i+j)%2==0)  
                s=s+a[i][j];  
        }  
    }  
    System.out.println(s);  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
s=0
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        if (i+j)%2==0:
```

```
            s=s+l[i][j]
```

```
print(s)
```

LBP246

Sum of elements whose sum of row index and col index is odd

Implement a program to find sum of row index and col index is odd in the given matrix.

input -----> a 3x3 matrix

con -----> no

output -----> sum as an int

formula: $(i+j)\%2==0$

logic

```
for(i=0;i<3;i++)
```

```
{
```

```
    for(j=0;j<3;j++)
```

```
    {
```

```
        if((i+j)%2==0)
```

```
            s=s+a[i][j];
```

```
    }
```

```
}
```

print s

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>

#include <stdlib.h>

int main() {

    int a[3][3],i,j,s;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    s=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            if((i+j)%2==0)

                s=s+a[i][j];

        }

    }

    printf("%d",s);

    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int a[][]=new int[3][3];
```

```
        int i,j,s=0;
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
            {
```

```
                a[i][j]=obj.nextInt();
```

```
            }
```

```
        }
```

```
        for(i=0;i<3;i++)
```

```
        {
```

```
            for(j=0;j<3;j++)
```

```
{  
    if((i+j)%2==0)  
        s=s+a[i][j];  
}  
}  
System.out.println(s);  
}  
}
```

python implementation:

```
l1=[int(i) for i in input().split()]  
l2=[int(i) for i in input().split()]  
l3=[int(i) for i in input().split()]  
l=[l1,l2,l3]  
s=0  
for i in range(3):  
    for j in range(3):  
        if (i+j)%2==0:  
            s=s+l[i][j]  
print(s)
```

LBP247

Sum of prime elements

Implement a program to find sum of prime elements in the given matrix.

input -----> a 3x3 matrix

con -----> no

output -----> sum as an int

LOGIC

```
s=0;
```

```
for(i=0;i<3;i++)
```

```
{
```

```
    for(j=0;j<3;j++)
```

```
    {
```

```
        if(isprime(a[i][j]))
```

```
            s=s+a[i][j];
```

```
    }
```

```
}
```

```
print s
```

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
int isprime(int n)  
{  
    int f=0,i;  
    for(i=1;i<=n;i++)  
    {  
        if(n%i==0)  
            f++;  
    }  
    return f==2;  
}  
  
int main() {  
    int a[3][3],i,j,s=0;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
}
```



```
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(isprime(a[i][j]))  
            s=s+a[i][j];  
    }  
}  
  
printf("%d",s);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    static boolean isprime(int n)  
  
    {  
  
        int i,f=0;
```

```
for(i=1;i<=n;i++)
{
    if(n%i==0)
        f++;
}
return f==2;
}

public static void main(String[] args) {
    Scanner obj = new Scanner(System.in);
    int a[][]=new int[3][3];
    int i,j,s=0;
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            a[i][j]=obj.nextInt();
        }
    }
    for(i=0;i<3;i++)
    {
        for(j=0;j<3;j++)
        {
            if(isprime(a[i][j]))
```

```
s=s+a[i][j];  
  
}  
  
}  
  
System.out.println(s);  
  
}  
  
}
```

python implementation:

```
def isprime(n):
```

```
    f=0
```

```
    for i in range(1,n+1):
```

```
        if n%i==0:
```

```
            f=f+1
```

```
    return f==2
```

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
s=0
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
if isprime(l[i][j]):  
    s=s+l[i][j]  
  
print(s)
```

LBP248

Count of prime digits in the given matrix

Implement a program to count number of prime digits present in the matrix.

input -----> a 3x3 matrix

con -----> no

output -----> prime digits count

logic:

```
count=0;
```

```
for(i=0;i<3;i++)
```

```
{
```

```
for(j=0;j<3;j++)  
{  
    count=count+countprime(a[i][j]);  
    123  
}  
}
```

```
int countprime(int n)  
{  
    while(n!=0)  
    {  
        d=n%10;  
        if(d==2 or 3 or 5 or 7)  
            c=c+1;  
        n=n/10;  
    }  
    return c;  
}
```

```
10 11 12  
13 14 15  
16 17 18  
0
```

10 ----> $0+0 = 0$

11 ----> $0+0 = 0$

12 ----> $0+1 = 1$

13 ----> $1+1 = 2$

14 ----> $2+0 = 2$

15 ----> $2+1 = 3$

16 ----> $3+0 = 3$

17 ----> $3+1 = 4$

18 ----> $4+0 = 4$

4

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int pc(int n)
```

```
{
```

```
    int c=0,d;
```

```
    while(n!=0)
```

```
{
```

```
d=n%10;

if(d==2 || d==3 || d==5 || d==7)

    c=c+1;

n=n/10;

}

return c;

}

int main() {

    int a[3][3],i,j,c=0;

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            scanf("%d",&a[i][j]);

        }

    }

    for(i=0;i<3;i++)

    {

        for(j=0;j<3;j++)

        {

            c=c+pc(a[i][j]);

        }

    }

}
```

```
printf("%d",c);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {  
  
    static int pc(int n)  
  
    {  
  
        int c=0,d;  
  
        while(n!=0)  
  
        {  
  
            d=n%10;  
  
            if(d==2 || d==3 || d==5 || d==7)  
  
                c++;  
  
            n=n/10;  
  
        }  
  
        return c;  
  
    }  
  
    public static void main(String[] args) {
```



```
Scanner obj = new Scanner(System.in);

int a[][]=new int[3][3];

int i,j,c=0;

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        a[i][j]=obj.nextInt();

    }

}

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        c=c+pc(a[i][j]);

    }

}

System.out.println(c);

}

}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
c=0
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        for k in str(l[i][j]):
```

```
            if k in "2357":
```

```
                c=c+1
```

```
print(c)
```

LBP249

Reverse of each element in matrix

Implement a program to reverse each element in the matrix.

input -----> a 3x3 matrix

con -----> no

output -----> result matrix

c implementation:

```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>  
  
int rev(int n)  
{  
    int r=0,d;  
    while(n!=0)  
    {  
        d=n%10;  
        r=r*10+d;  
        n=n/10;  
    }  
    return r;  
}  
  
int main() {  
    int a[3][3],i,j;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            scanf("%d",&a[i][j]);  
        }  
    }  
}
```

```
}  
  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        printf("%d ",rev(a[i][j]));  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
int a[][]= new int[3][3];

int i,j;

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        a[i][j]=obj.nextInt();

    }

}

for(i=0;i<3;i++)

{

    for(j=0;j<3;j++)

    {

        //convert into string

        //string into string buffer

        //cal reverse()

        System.out.print(new StringBuffer(Integer.toString(a[i][j])).reverse()+"

");

    }

    System.out.println();

}

}
```

python implementation:

```
l1=[int(i) for i in input().split()]
```

```
l2=[int(i) for i in input().split()]
```

```
l3=[int(i) for i in input().split()]
```

```
l=[l1,l2,l3]
```

```
for i in range(3):
```

```
    for j in range(3):
```

```
        print(str(l[i][j])[::-1],end=' ')
```

```
    print()
```

LBP250

Keep palindromic number and replace remaining with 0's

Implement a program to keep all palindromic numbers as it is and replace remaining with 0.

input -----> a 3x3 matrix

con -----> no

output -----> result matrix

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int rev(int n)
```

```
{
```

```
    int r=0,d;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        r=r*10+d;
```

```
        n=n/10;
```

```
    }
```

```
    return r;
```

```
}
```

```
int main() {
```

```
    int a[3][3],i,j;
```

```
    for(i=0;i<3;i++)
```

```
    {
```

```
        for(j=0;j<3;j++)
```

```
        {
```

```
scanf("%d",&a[i][j]);  
  
}  
  
}  
  
for(i=0;i<3;i++)  
{  
    for(j=0;j<3;j++)  
    {  
        if(a[i][j]==rev(a[i][j]))  
            printf("%d ",a[i][j]);  
        else  
            printf("0 ");  
    }  
    printf("\n");  
}  
  
return 0;  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
  
public class Solution {
```



```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int a[][]= new int[3][3];  
    int i,j;  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            a[i][j]=obj.nextInt();  
        }  
    }  
    for(i=0;i<3;i++)  
    {  
        for(j=0;j<3;j++)  
        {  
            //convert into string  
            //string into string buffer  
            //cal reverse()  
            String s1 = Integer.toString(a[i][j]);  
            String s2 = new StringBuffer(s1).reverse().toString();  
            if(s1.equals(s2))  
                System.out.print(a[i][j]+" ");  
        }  
    }  
}
```

```
        else

            System.out.print("0 ");

        }

        System.out.println();

    }

}

}
```

python implementation:

```
l1=[int(i) for i in input().split()]
l2=[int(i) for i in input().split()]
l3=[int(i) for i in input().split()]
l=[l1,l2,l3]

for i in range(3):

    for j in range(3):

        s=str(l[i][j])

        if s==s[::-1]:

            print(l[i][j],end=' ')

        else:

            print('0',end=' ')

    print()
```

LBP251

multiples of 10

Given an array A of N integer numbers.

The task is to rewrite the array by putting all multiples of 10 at the end of the given array.

Note: The order of the numbers which are not multiples of 10 should remain unaltered, and similarly. the order of all multiples of 10 should be unaltered.

input -----> N and Array Elements

con -----> no

output -----> updated array

10 11 20 15 30 45 50 ==> 11 15 45 10 20 30 50

if(a[i]%10!=0) print

if(a[i]%10==0) print

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int n,a[100],i;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
    {  
        if(a[i]%10!=0)  
            printf("%d ",a[i]);  
    }  
    for(i=0;i<n;i++)  
    {  
        if(a[i]%10==0)  
            printf("%d ",a[i]);  
    }  
    return 0;  
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt(),i;

        int a[]=new int[n];

        for(i=0;i<n;i++)

            a[i]=obj.nextInt();

        for(i=0;i<n;i++)

        {

            if(a[i]%10!=0)

                System.out.print(a[i]+" ");

        }

        for(i=0;i<n;i++)

        {

            if(a[i]%10==0)

                System.out.print(a[i]+" ");

        }

    }

}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
for i in L:
```

```
    if i%10!=0:
```

```
        print(i,end=' ')
```

```
for i in L:
```

```
    if i%10==0:
```

```
        print(i,end=' ')
```

LBP252

Employee's Rating Point

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 -----> an integer value

con -----> con

output -----> employee rating grade

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

```
    scanf("%d",&n);
```

```
    if(n>=30&& n<=50)
```

```
        printf("D");
```

```
    else if(n>=51&& n<=60)
```

```
        printf("C");
```

```
else if(n>=61&& n<=80)

    printf("B");

else

    printf("A");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        if(n>=30 && n<=50)

            System.out.println("D");

        else if(n>=51 && n<=60)

            System.out.println("C");

        else if(n>=61 && n<=80)

            System.out.println("B");

    }

}
```


else

```
System.out.println("A");
```

```
}
```

```
}
```

python implementation:

```
n=int(input())
```

```
if n>=30 and n<=50:
```

```
    print("D")
```

```
elif n>=51 and n<=60:
```

```
    print("C")
```

```
elif n>=61 and n<=80:
```

```
    print("B")
```

```
else:
```

```
    print("A")
```

LBP253

encrypted digits

A company transfers an encrypted code to one of its clients.

The code needed to be decrypted so that it can be used for accessing all the required information.

The code can be decrypted by interchanging each consecutive digit and once if the digit got interchanged then it cannot be used again.

If at a certain point there is no digits to be interchanged with, then that single digit must be left as it is.

Write an algorithm to decrypt the code so that it can be used to access the required information.

input -----> a number from the user

con -----> no

output -----> an integer value

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,i,a[100];
```

```
    scanf("%d",&n);
```

```
for(i=0;i<n;i++)
    scanf("%d",&a[i]);
if(n%2==0)
{
    for(i=0;i<n;i=i+2)
    {
        printf("%d %d ",a[i+1],a[i]);
    }
}
else{
    for(i=0;i<n-1;i=i+2)
        printf("%d %d ",a[i+1],a[i]);
    printf("%d ",a[i]);
}
return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;
```

```
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int i,n=obj.nextInt();  
    int a[]=new int[n];  
    for(i=0;i<n;i++)  
        a[i]=obj.nextInt();  
    if(n%2==0)  
    {  
        for(i=0;i<n;i=i+2)  
            System.out.print(a[i+1]+" "+a[i]+" ");  
    }  
    else{  
        for(i=0;i<n-1;i=i+2)  
            System.out.print(a[i+1]+" "+a[i]+" ");  
        System.out.print(a[i]);  
    }  
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
if n%2==0:
```

```
    i=0
```

```
    while i<n:
```

```
        print(L[i+1],L[i],end=' ')
```

```
        i=i+2
```

```
else:
```

```
    i=0
```

```
    while i<n-1:
```

```
        print(L[i+1],L[i],end=' ')
```

```
        i=i+2
```

```
    print(L[n-1])
```

LBP254

Easy Shopping

The e-commerce company 'Easy Shopping' displays some specific products for its premium customers on its user interface.

The company shortlisted a list of N products.

The list contains the price of each product.

The company will select random products for display.

The selection criteria states that only those products whose price is a perfect cube number will be selected for display.

The selection process is automated and is done by the company's system.

The company wishes to know the total count of the products selected for display.

Write an algorithm to find the count of products that will be displayed.

input ----> an array size and elements

con -----> no

output ---> perfect cube elements

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int i,a[100],n,c;
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)
```

```
        scanf("%d",&a[i]);
```

```
    for(i=0;i<n;i++)
```

```
{  
  
    c=0;  
  
    while(c<=a[i])  
  
    {  
  
        if(c*c*c==a[i])  
  
        {  
  
            printf("%d ",a[i]);  
  
            break;  
  
        }  
  
        c++;  
  
    }  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
Scanner obj = new Scanner(System.in);

int n=obj.nextInt();

int i,c;

int a[]=new int[n];

for(i=0;i<n;i++)

    a[i]=obj.nextInt();

for(i=0;i<n;i++)

{

    c=0;

    while(c<=a[i])

    {

        if(c*c*c==a[i])

        {

            System.out.print(a[i]+" ");

            break;

        }

        c++;

    }

}

}
```

python implementation:


```
-----  
n=int(input())  
L=[int(i) for i in input().split()]  
for i in L:  
    c=0  
    while c<=i:  
        if c*c*c==i:  
            print(i,end=' ')  
            break  
        c=c+1
```

LBP255

player's score

In a game, organizers has given a number to the player.

The player has to find out the differnece between the number and the reverse of the number.

The difference between two numbers is the player's score.

The number given to the player and the player's score can be a negative or positive number.

Write an algorithm to find the player's score.

input -----> an integer

con -----> no

output -----> player's score

n-rev(n)

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int rev(int n)
```

```
{
```

```
    int d;
```

```
    int r=0;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        r=r*10+d;
```

```
        n=n/10;
```

```
    }
```

```
    return r;
```

```
}  
  
int main() {  
    int n;  
    scanf("%d",&n);  
    printf("%d",n-rev(n));  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
    static int rev(int n)  
    {  
        int d;  
        int r=0;  
        while(n!=0)  
        {  
            d=n%10;  
            r=r*10+d;  
            n=n/10;  
        }  
    }  
}
```

```
}  
  
    return r;  
  
}  
  
public static void main(String[] args) {  
  
    Scanner obj = new Scanner(System.in);  
  
    int n=obj.nextInt();  
  
    System.out.println(n-rev(n));  
  
}  
}
```

python implementation:

```
n=input()  
print(int(n)-int(n[::-1]))
```

LBP256

GlobalAdd

The media compnay "GlobalAdd" has received a batch of advertisements from different product brands.

The batch of advertisements is a numeric value where each digit represnets the number of advertisements the media company has received from different product brands.

Since the company banners permit only even numbers of advertisements to be displayed,

the media company needs to know the total number of advertisements it will be able to display from the given batch.

Write an algorithm to calculate the total number of advertisements that will be displayed from the batch.

input ----> an integer

con -----> no

output ----> count of advertisements

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,c=0,d;
```

```
    scanf("%d",&n);
```

```
    while(n!=0)
```

```
{
```

```
d=n%10;

if(d%2==0)

    c++;

n=n/10;

}

printf("%d",c);

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int d,c=0;

        while(n!=0)

        {

            d=n%10;
```

```
        if(d%2==0)

            c++;

            n=n/10;

        }

        System.out.println(c);

    }

}
```

python implementation:

```
n=int(input())

c=0

while n!=0:

    d=n%10

    if d%2==0:

        c=c+1

    n=n//10

print(c)
```

LBP257

FunGames

The games development company "FunGames" has developed a ballon shooter games.

The ballons are arranged in a linear sequence and each ballon has a number associated with it.

The numbers on the ballons are fibonacci series.

In the game the player shoots 'k' ballons.

The player's score is the sum of numbers on k ballons.

Write an algorithm to generate the player's score.

input ----> an integer vlaue n

con -----> no

output ---> sum value

0 1 1 2 3 5 8

K=1 ----> 0

K=3 ----> 0+1+1=2

sum of fib seq

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int i,k,a1,a2,a3,sum;
```

```
    scanf("%d",&k);
```

```
    a1=-1;
```

```
    a2=1;
```

```
    sum=0;
```

```
    for(i=1;i<=k;i++)
```

```
    {
```

```
        a3=a1+a2;
```

```
        a1=a2;
```

```
        a2=a3;
```

```
        sum=sum+a3;
```

```
    }
```

```
    printf("%d",sum);
```

```
    return 0;
```

```
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int k=obj.nextInt();  
  
        int i,sum=0,a1=-1,a2=1,a3;  
  
        for(i=1;i<=k;i++)  
        {  
            a3=a1+a2;  
  
            sum=sum+a3;  
  
            a1=a2;  
  
            a2=a3;  
  
        }  
  
        System.out.println(sum);  
    }  
}
```

python implementation:

```
k=int(input())  
  
a1=-1  
  
a2=1  
  
sum=0  
  
for i in range(1,k+1):  
    a3=a1+a2  
    sum=sum+a3  
    a1,a2=a2,a3  
  
print(sum)
```

LBP258

The Past Book

To create a profile on a social media account "ThePastBook".

The user needs to enter a string value in the form of user name.

The username should consist of only characters tagged a-z.

if the user enters an incorrect string containing digits the system automatically identifies the number of digits in the string and removes them.

Write an algorithm to help the system identify the count of digits present in the username.

input -----> A string from the user.

con -----> no

output ----> count of digits

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,c=0;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        if(s[i]>='0'&& s[i]<='9')
```

```
            c++;
```

```
    }
```

```
    printf("%d",c);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        String s = obj.nextLine();
```

```
        int i,c=0;
```

```
        for(i=0;i<s.length();i++)
```

```
        {
```

```
            if(s.charAt(i)>='0' && s.charAt(i)<='9')
```

```
                c++;
```

```
        }
```

```
        System.out.println(c);
```

```
    }
```

```
}
```

python implementation:

```
s=input()
```

```
c=0
```

```
for i in s:
```

```
    if i.isdigit():
```

```
        c=c+1
```

```
print(c)
```

LBP259

Morning Prayer

Students of a school are assembled in a straight line for the morning prayer.

To uplift the spirit of the students, an exercise is conducted.

The initial letter of all the student's names is noted down(str).

The task here is to substitute the initial letters in the list with consonants

such that if the initial letter of the student is a vowel, retain the student in the same position.

If the initial letter of the student is a vowel, swap with the next immediate consonants from the english alphabet sequence (a-z).

Say, initial letter of a student starts with b swap with c, next immediate consonant.

If the initial letter is 'z' swap with 'b'.

input ----> a string from the user

con -----> no

output ----> updated string

welcome -----> xemdone

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100],ch;
```

```
    int i;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++)
```

```
    {
```

```
        if(s[i]=='a' || s[i]=='e' || s[i]=='i' || s[i]=='o' || s[i]=='u')
```

```
            printf("%c",s[i]);
```

```
        else
```

```
        {
```

```
            ch = s[i];
```

```
if(ch=='z')

    ch = 'a';

ch++;

if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')

    ch++;

printf("%c",ch);

}

}

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        String s = obj.nextLine();
```

```
        int i;
```

```
        char ch;
```



```
for(i=0;i<s.length();i++)
{
    ch=s.charAt(i);
    if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
        System.out.print(ch);
    else
    {
        char ch1=ch;
        if(ch1=='z')
            ch1='a';
        ch1++;
        if(ch1=='a' || ch1=='e' || ch1=='i' || ch1=='o' || ch1=='u')
            ch1++;
        System.out.print(ch1);
    }
}
```

python implementation:

```
s=input()
```

```
for i in s:
```

if i in 'aeiou':

```
print(i,end="")
```

else:

```
ch = i
```

```
if ch=='z':
```

```
ch='a'
```

```
ch=chr(ord(ch)+1)
```

```
if ch in 'aeiou':
```

```
print(chr(ord(ch)+1),end="")
```

```
else:
```

```
print(ch,end="")
```

LBP260

factorial game

Mikes likes to play with numbers.

His friends are also good with numbers and often plays mathematical games.

they made a small game where they will spell the last digit of a factorial of a number other than 0.

Let say the given number is 5, so 5! will be 120, Here 0 is the last digit.

But we dn't want 0, we want a number other than 0. Then last digit is 2.

input ----> an integer value

con -----> no

output ---> an integer vlaue

0!=1 -----> 1

1!=1 -----> 1

2!=2 -----> 2

3!=6 -----> 6

4!=24 ----> 4

5!=120 ---> 2

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,i,f;
```

```
    scanf("%d",&n);
```

```
    f=1;
```

```
for(i=1;i<=n;i++)  
    f=f*i;  
if(f%10==0)  
    printf("%d",(f/10)%10);  
else  
    printf("%d",f%10);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();  
        int i,f=1;  
        for(i=1;i<=n;i++)  
            f=f*i;  
        if(f%10==0)
```

```
        System.out.println((f/10)%10);  
  
    else  
  
        System.out.println(f%10);  
  
    }  
  
}
```

python implementation:

```
-----  
  
import math  
  
n=int(input())  
  
f=math.factorial(n)  
  
if f%10==0:  
  
    print((f//10)%10)  
  
else:  
  
    print(f%10)
```

LBP261

Speed Maths

Jack was in 9th standard. He appeared for a speed maths competitive exam.

Jack is taking longer time to solve one of the problems.

Count the number of 1's in the binary representation of an integer.

Help him to solve the below problem and write a code for the same.

input -----> an integer value

con -----> no

output -----> an int value

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d,c;
```

```
    scanf("%d",&n);
```

```
    c=0;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%2;
```

```
        if(d==1)
```

```
            c++;
```

```
        n=n/2;
```

```
}  
  
printf("%d",c);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),c=0,d;  
  
        while(n!=0)  
        {  
            d=n%2;  
  
            if(d==1)  
                c=c+1;  
  
            n=n/2;  
        }  
  
        System.out.println(c);  
    }  
}
```

```
}  
  
}
```

python implementation:

```
n=int(input())  
s=bin(n)  
print(s.count('1'))
```

LBP262

puzzle

Dennis was solving a puzzle.

the puzzle was to verify a number whose cube ends with the number itself.

Help Dennis to find the solution of the puzzle and write the code accordingly.

input -----> integer value to verified

con -----> no

output -----> boolean value True or False

c implementation:


```
-----  
  
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>
```

```
  
int main() {  
  
    int n;  
  
    scanf("%d",&n);  
  
    if(n*n*n%10==n)  
        printf("true");  
  
    else  
        printf("false");  
  
    return 0;  
  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
  
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n = obj.nextInt();  
    System.out.println(n*n*n%10==n);  
}  
}
```

python implementation:

```
-----  
n=int(input())  
print('true' if n**3%10==n else 'false')
```

LBP263

mathematics class

In a mathematics class, number system is being taught to students, before teaching them 10's and 100's place, they will be taught the number positions.

The positions will be starting from sequence number 1 and the direction will be from left to right.

So if i want to find second position of a digit in the number 90876, it will be 0.

if the kth digit exceeds the number position return -1.

write a program to find the kth digit in a given number.

input ----> integer value and kth digit

con -----> no

output ----> an integer number

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i;
```

```
    scanf("%s",s);
```

```
    scanf("%d",&i);
```

```
    if(i<strlen(s))
```

```
        printf("%c",s[i-1]);
```

```
    else
```

```
printf("-1");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        int i=obj.nextInt();  
  
        System.out.println((i<s.length())?s.charAt(i-1):"-1");  
  
    }  
  
}
```

python implementation:

```
s=input()  
  
n=int(input())
```

```
print(s[n-1] if n<len(s) else '-1')
```

```
(i<s.length())?s.charAt(i-1):"-1"
```

abc, 1 ---> a

abc, 2 ---> b

abc, 3 ---> c

abc, 4 ---> -1

LBP264

power function

In a mathematics class, the students are being taught power function.

So "a" raised to the power of "b" is shown as a^b and the calculation goes as $a*a*a...b$ times.

Now there is slight twist to the problem,

the students have to find out the last digit of the resultant a^b .

input -----> an integer value as base and power

con -----> no

output -----> last digit of a^b

4,2 ---> 16 ---> 6

2,10 --> 1024 -> 4

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int a,b;

    scanf("%d %d",&a,&b);

    printf("%d",((int)pow(a,b))%10);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int a=obj.nextInt();  
  
        int b = obj.nextInt();  
  
        System.out.println(((int)Math.pow(a,b))%10);  
  
    }  
}
```

python implementation:

```
a,b=(int(i) for i in input().split())  
print((a**b)%10)
```

LBP265

mathematical tricks

Aryan is studying in the 5th standard.

He is very interested in mathematical tricks and always wanted to play with numbers.

Aryan would like to replace existing numbers with some other numbers.

Today he decided to replace all digits of the number

(which is greater than or equal to 2 digits) by its squares and print it in reverse order.

Write a program to help him to replace numbers accordingly.

input -----> a number

con -----> no

output ----> replaced all digits by its squares and reversed number

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d;
```

```
    scanf("%d",&n);
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```



```
printf("%d",d*d);  
  
n=n/10;  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int d;  
  
        while(n!=0)  
  
        {  
  
            d=n%10;  
  
            System.out.print(d*d);  
  
            n=n/10;
```

```
}  
  
}  
  
}
```

python implementation:

```
n=int(input())  
while n!=0:  
    d=n%10  
    print(d*d,end="")  
    n=n//10
```

LBP266

coding standards

Tom has joined a new company and he is assigned a program to code.

But before starting to code, he needs to know the coding standards.

He need to make sure that the variable name should meet the below standards,

=> contains only english letter

=> and/or digits

=> and/or underscore (_)

=> should not start with digits

The program should return True/False based on the above conditions

input ----> a string from the user

con -----> no

output ---> true or false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,flag=1;
```

```
    scanf("%s",s);
```

```
    for(i=0;s[i];i++)
```

```
{
```

```
if((s[i]>='a'&& s[i]<='z') || (s[i]>='A'&& s[i]<='Z') || (s[i]=='_') || (s[i]>='0'&& s[i]<='9'
))
{
    if(s[0]>='0'&& s[0]<='9')
    {
        flag=0;
        break;
    }
}
else
{
    flag=0;
    break;
}
}
printf((flag==1)?"true":"false");
return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        String s = obj.nextLine();
```

```
        System.out.println(s.matches("[a-zA-Z_][a-zA-Z0-9_]+"));
```

```
    }
```

```
}
```

python implementation:

```
import re
```

```
s=input()
```

```
print('true' if re.fullmatch("[a-zA-Z_][a-zA-Z0-9_]+",s) else 'false')
```

```
[a-zA-Z_][a-zA-Z0-9_][a-zA-Z0-9_]+
```

```
abc
```

```
a
```

LBP267

party quiz

While sitting in party, Tom came up with an idea of a quiz.

and the quiz is, Tom will spell out a number, and a person has to tell a number which is next to it. But this number has to be perfect square.

input -----> a number from the user

con -----> no

output ----> the perfect square after N

n=7 ---> 8, 9, 10, 11, 12, ----> 9

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
int n,i;

scanf("%d",&n);

for(i=1;;i++)
{
    if(i*i>=n){
        printf("%d",i*i);
        break;
    }
}

return 0;
}
```

java implementation:

```
-----

import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();
```

```
int i;  
  
for(i=1;;i++)  
{  
    if(i*i>=n)  
    {  
        System.out.print(i*i);  
        break;  
    }  
}  
}
```

python implementation:

```
n=int(input())  
  
i=1  
  
while True:  
    if i*i>=n:  
        print(i*i)  
        break  
  
    i=i+1
```

LBP268

Be Positive

Write a program to get two inputs from the user and find the absolute difference between the sum of two numbers and the product of two numbers.

input -----> two numbers from the user

con -----> no

output -----> absolute difference

a and b

$\text{abs}((a+b)-(a*b))$

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a,b;
```

```
scanf("%d %d",&a,&b);  
  
printf("%d",abs((a+b)-(a*b)));  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int a=obj.nextInt();  
  
        int b=obj.nextInt();  
  
        System.out.println(Math.abs((a+b)-(a*b)));  
  
    }  
  
}
```

python implementation:

```
a=int(input())
```

```
b=int(input())  
print(abs((a+b)-(a*b)))
```

LBP269

Prime Number Busses

James wants to travel by bus to reach his friend John's home.

John gave a hint that all busses from Jame's location will reach his home
if the bus number is prime number.

Write a program to help James find the bus that reaches John's home.

input -----> a number from the user

con -----> no

output ----> yes or no

c implementation:

```
#include <stdio.h>  
  
#include <string.h>  
  
#include <math.h>  
  
#include <stdlib.h>
```

```
int main() {  
    int n,i,f;  
    scanf("%d",&n);  
    f=0;  
    for(i=1;i<=n;i++)  
    {  
        if(n%i==0)  
            f++;  
    }  
    printf((f==2)?"yes":"no");  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),i,f;
```

```
f=0;

for(i=1;i<=n;i++)
{
    if(n%i==0)
        f++;
}

System.out.println((f==2)?"yes":"no");
}
```

python implementation:

```
-----

n=int(input())

f=0

for i in range(1,n+1):

    if n%i==0:

        f=f+1

print('yes' if f==2 else 'no')
```

LBP270

sentence making

The teacher in a school has started to teach the very basics rule for a sentence is that it should start with a capital letter and ends with a full stop.

If the sentence fails any of the above mentioned criteria, it will be an incorrect sentence.

Make a program to validate whether a given statement is a correct sentence or not.

The program should return True/False based on the validity.

input ----> a string from the user

con -----> no

output ----> true or false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    scanf("%[^\\n]s",s);
```

```
    if((s[0]>='A'&& s[0]<='Z')&&(s[strlen(s)-1]=='.'))
```

```
        printf("true");
```

```
else  
  
    printf("false");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.nextLine();  
  
        System.out.println(s.matches("[A-Z][a-zA-Z_0-9 ]*"));  
  
    }  
  
}
```

* zero or more characters

+ one or more characters

* ch ---> 0 or 1 or 2 or 3 and son on ---> atleast

+ ch ---> 1 or 2 or 3 and son on -----> atmost

python implementation:

```
import re
```

```
s=input()
```

```
print('true' if re.fullmatch("[A-Z][a-zA-Z_0-9 ]*.[.]",s) else 'false')
```

LBP271

Single Binary Value

Geetha Singh is trying to create a system to convert binary number to its decimal equivalent. Help her to automate the process.

input -----> a binary value

con-----> no

output ----> decimal value

8421

n=110 --->4+2=6

$$0 \times 2^0 + 1 \times 2^1 + 1 \times 2^2$$

$$0 + 2 + 4 = 6$$

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,dec=0,d,i=0;
```

```
    scanf("%d",&n);
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        dec=dec+d*(int)pow(2,i++);
```

```
        n=n/10;
```

```
    }
```

```
    printf("%d",dec);
```

```
    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        String s = obj.nextLine();
```

```
        System.out.println(Integer.parseInt(s,2));
```

```
    }
```

```
}
```

python implementation:

```
print(eval("0b"+input()))
```

LBP272

Item id

A company wishes to bucketize their item id's 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 ----> ItemId

con -----> no

output ----> bucket ID

12875 ---> 8

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,d,m;
```

```
    scanf("%d",&n);
```

```
    m=0;
```

```
    while(n!=0)
```

```
{  
    d=n%10;  
    if(d>m)  
        m=d;  
    n=n/10;  
}  
printf("%d",m);  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt(),d,m;  
        m=0;  
        while(n!=0)
```

```
{  
    d=n%10;  
    if(d>m)  
        m=d;  
    n=n/10;  
}  
System.out.println(m);  
}
```

python implementation:

```
print(max([int(i) for i in input()])))
```

LBP273

Next Letter

Implement the following function

```
def NextLetter(ch1,ch2);
```

The function accepts two characters ch1 and ch2 as arguments, ch1 and ch2 are alphabetical letters.

Implement the function to find and return the next letter so that distance between ch2 and the next letter is the same as the distance between ch1 and ch2.

While counting distance if you exceed the letter 'z' then, count the remaining distance starting from the letter 'a'.

Distance between two letters is the number of letters between them.

input -----> char ch1 and char ch2

con -----> no

output -----> char ch

$ch2 + (ch2 - ch1)$

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    char ch1,ch2;  
    scanf("%c %c",&ch1,&ch2);  
    int d = ch2-ch1;  
    if(ch2=='z')  
        printf("%c",'a'+d);  
    else  
        printf("%c",ch2+d);  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        char ch1 = obj.next().charAt(0);  
        char ch2 = obj.next().charAt(0);
```

```
int d = ch2-ch1;

if(ch2=='z')

    System.out.println((char)('a'+d));

else

    System.out.println((char)(ch2+d));

}

}
```

python implementation:

```
-----

ch1,ch2=(i for i in input().split())

d=ord(ch2)-ord(ch1)

if ch2=='z':

    print(chr(ord('a')+d))

else:

    print(chr(ord(ch2)+d))
```

LBP274

Sum of All Integers

Implement a program to find sum of all integers between two integer numbers taken as input and are divisible by 7.

input -----> an integer value

con -----> no

output -----> sum value

sum of numbers which are divisible by 7

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int n1,n2,i,sum=0;
    scanf("%d %d",&n1,&n2);
    for(i=n1;i<=n2;i++)
    {
        if(i%7==0)
```

```
        sum=sum+i;

    }

    printf("%d",sum);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n1=obj.nextInt();

        int n2=obj.nextInt();

        int i,sum=0;

        for(i=n1;i<=n2;i++)

        {

            if(i%7==0)

                sum=sum+i;

        }

    }

}
```

```
System.out.println(sum);  
  
}  
  
}
```

python implementation:

```
-----  
  
n1,n2=(int(i) for i in input().split())  
  
sum=0  
  
for i in range(n1,n2+1):  
  
    if i%7==0:  
  
        sum=sum+i  
  
print(sum)
```

LBP275

Country Visa Center

The country visa center generates the token number for its applicants from their application ID.

the application ID is numeric value.

The token number is generated in a specific form.

the even digits in the applicant's ID are replaced by the digit one greater than the even digit

and the odd digits in the applicant's ID are replaced by the digit one lesser than the odd digit.

The numeric value thus generated represents the token number of applicant.

Write an algorithm to generate the token number from the applicant ID.

input ----> application ID

con -----> no

output ----> token id

12345 ---> 03254

add 1 to even digits

sub 1 from odd digits

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int rev(int n)
```

```
{
```

```
int r=0,d;

while(n!=0)

{

    d=n%10;

    r=r*10+d;

    n=n/10;

}

return r;

}

int main() {

    int n,d;

    scanf("%d",&n);

    n=rev(n);

    while(n!=0)

    {

        d=n%10;

        if(d%2==0)

            printf("%d",d+1);

        else

            printf("%d",d-1);

        n=n/10;

    }

    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    static int rev(int n)
```

```
    {
```

```
        int r=0,d;
```

```
        while(n!=0)
```

```
        {
```

```
            d=n%10;
```

```
            r=r*10+d;
```

```
            n=n/10;
```

```
        }
```

```
        return r;
```

```
    }
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int n=obj.nextInt();
```

```
        int d;
```

```
n=rev(n);  
while(n!=0)  
{  
    d=n%10;  
    if(d%2==0)  
        System.out.print(d+1);  
    else  
        System.out.print(d-1);  
    n=n/10;  
}  
}
```

python implementation:

```
n=input()  
n=int(n[::-1])  
while n!=0:  
    d=n%10  
    if d%2==0:  
        print(d+1,end="")  
    else:  
        print(d-1,end="")
```

$n = n // 10$

LBP276

Neon Number

Rahul's teacher gave an assignment to all the student that when they show up tomorrow they should find a special type of number and report her.

He thought very carefully and came up with an idea to have neon numbers.

A neon number is a number where the square of the sum of each digit of the number results in the given number.

Given an integer N,

Write a program to find whether the number N is a Neon number.

If it's not a Neon number, print the square of the sum of digits of the number.

input -----> a number

con -----> no

output -----> true or false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```



```
#include <stdlib.h>
```

```
int main() {  
    int n,a,b,c;  
    scanf("%d",&n);  
    a=n%10;  
    b=(n/10)%10;  
    c=(a+b)*(a+b);  
    printf((c==n)?"true":"false");  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();
```

```
int a,b,c;  
  
a=n%10;  
  
b=(n/10)%10;  
  
c=(a+b)*(a+b);  
  
System.out.println(c==n);  
  
}  
  
}
```

python implementation:

```
n=int(input())  
  
a=n%10  
  
b=(n//10)%10  
  
print("true" if (a+b)**2==n else "false")
```

LBP277

super market pricing

A super market maintains a pricing format for all its products. A value N is printed on each product. When the scanner reads the value N on the item, the product of all the digits in the value N is the price of the item. the task here is to design the software such that given the code of any item N the product (multiplication) of all the digits of value should be computed (price).

input ----> an integer value

con -----> no

output ---> price

123=1*2*3=6/-

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,prod=1,d;
```

```
    scanf("%d",&n);
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        prod=prod*d;
```

```
        n=n/10;
```

```
}  
  
printf("%d",prod);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int prod=1,d;  
  
        while(n!=0)  
        {  
            d=n%10;  
  
            prod=prod*d;  
  
            n=n/10;  
        }  
  
        System.out.println(prod);  
    }  
}
```

```
}  
  
}
```

python implementation:

```
import math  
  
L=[int(i) for i in input()]  
  
print(math.prod(L))
```

LBP278

Number Container

Given two positive numbers N and K. The task is to find the number N contains exactly K digit or not. If contains then print True<space>digit_count otherwise False<space>digit_count.

input ----> Value of N and K

con -----> con

output ---> True|False<space>Digit_Count

123 4 ----> False 3

124 3 ----> True 3

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char n1[100];

    int n2;

    scanf("%s",n1);

    scanf("%d",&n2);

    if(strlen(n1)==n2)

        printf("True %d",strlen(n1));

    else

        printf("False %d",strlen(n1));

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        String s = obj.next();  
  
        int n = obj.nextInt();  
  
        if(s.length()==n)  
            System.out.println("True "+s.length());  
        else  
            System.out.println("False "+s.length());  
    }  
}
```

python implementation:

```
-----  
n1,n2=(i for i in input().split())  
n2=int(n2)  
if len(n1)==n2:  
    print("True",len(n1))  
else:  
    print("False",len(n1))
```

LBP279

pronic number

A pronic number is a number which is a product of two consecutive integers, that is,

a number of the form $n(n + 1)$.

Create a function that determines whether a number is pronic or not.

input-----> a number from the user

con -----> no

output ----> true or false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,x;
```



```
scanf("%d",&n);  
  
x=(int)sqrt(n);  
  
if(n==x*(x+1))  
    printf("true");  
  
else  
    printf("false");  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int x=(int)Math.sqrt(n);  
  
        System.out.println(x*(x+1)==n);  
  
    }  
  
}
```

python implementation:

```
import math
n=int(input())
x=math.isqrt(n)
print("true" if n==x*(x+1) else "false")
```

LBP280

Validate ATM PIN

Implement a program that will test if a string is a valid PIN or not via a regular expression.

A valid PIN has:

Exactly 4 characters.

Only numeric characters (0-9).

No whitespace.

input ----> an input from the user

con -----> con

output ---> true or false

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    char s[100];

    scanf("%s",s);

    int i,c=0;

    for(i=0;s[i];i++)

    {

        if(s[i]>='0'&& s[i]<='9')

            c++;

    }

    printf((c==4)? "true": "false");

    return 0;

}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        System.out.println(s.matches("[0-9]{4}"));  
    }  
}
```

python implementation:

```
-----  
  
import re  
  
s=input()  
  
print('true' if re.fullmatch("[0-9]{4}",s)!=None else 'false')
```

LBP281

The Actual Memory Size of Your USB Flash Drive

Implement a program that takes the memory size as an argument and returns the actual memory size.

Note: The actual storage loss on a USB device is 7% of the overall memory size!

input -----> memory size in GB

con -----> no

output -----> actual memory size, round your result to two decimal places.

1GB -----> 0.93GB

$n - n * 7.0 / 100$ ---> actual size

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

```
scanf("%d",&n);

printf("%.2f",n-n*(7.0/100));

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n = obj.nextInt();

        System.out.printf("%.2f",n-n*0.07);

    }

}
```

python implementation:

```
n=int(input())

print("%.2f"%(n-n*0.07))
```

LBP282

Happy Number

The happy number can be defined as a number which will yield 1 when it is replaced by the sum of the square of its digits repeatedly.

If this process results in an endless cycle of numbers containing 4, then the number is called an unhappy number.

Write a program that accepts a number and determines whether the number is a happy number or not. Return true if so, false otherwise.

input -----> a number from the user

con -----> no

output ----> true or false

$32=9+4=13=1+9=10=1+0=1$ true

$16=1+36=37=9+49=58=25+64=89=.....=4$ false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>

#include <stdlib.h>

int sum(int n)
{
    int d,s=0;
    while(n!=0)
    {
        d=n%10;
        s=s+d*d;
        n=n/10;
    }
    return s;
}

int main() {
    int n;
    scanf("%d",&n);
    while(1)
    {
        if(n==1)
        {
            printf("true");
            break;
        }
    }
```



```
if(n==4)
{
    printf("false");
    break;
}
n=sum(n);
}
return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;
```

```
public class Solution {
    static int sum(int n)
    {
        int d,s=0;
        while(n!=0)
        {
            d=n%10;
            s=s+d*d;
        }
    }
}
```

```
n=n/10;

}

return s;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n=obj.nextInt();

    while(true)

    {

        if(n==1)

        {

            System.out.println(true);

            break;

        }

        if(n==4)

        {

            System.out.println(false);

            break;

        }

        n=sum(n);

    }

}

}
```

python implementation:

```
def sum(n):
```

```
    s=0
```

```
    while n!=0:
```

```
        d=n%10
```

```
        s=s+d*d
```

```
        n=n//10
```

```
    return s
```

```
n=int(input())
```

```
while True:
```

```
    if n==1:
```

```
        print('true')
```

```
        break
```

```
    if n==4:
```

```
        print('false')
```

```
        break
```

```
    n=sum(n)
```

LBP283

Calculate the Mean

Implement a function that takes an array of numbers and returns the mean (average) of all those numbers.

input ----> an array size and elements

con -----> no

output ---> print mean value and round to two decimal places.

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int n,a[100],i,sum;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    sum=0;
    for(i=0;i<n;i++)
        sum=sum+a[i];
```

```
printf("%.2f",(float)sum/n);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int sum=0,i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
            sum=sum+a[i];  
  
        System.out.printf("%.2f",(double)sum/n);  
  
    }  
  
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
sum=0
```

```
for i in L:
```

```
    sum=sum+i
```

```
print("%.2f"%(sum/n))
```

LBP284

Factorize a Number

Implement a program to that takes a number as its argument and returns an array of all its factors

input -----> a number

con -----> no

output ----> list of factors

```
for(i=1;i<=n;i++)
```

```
{
```

```
        if(n%i==0)
            print i
    }
```

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int n,i;
    scanf("%d",&n);
    for(i=1;i<=n;i++)
    {
        if(n%i==0)
            printf("%d ",i);
    }
    return 0;
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        for(i=1;i<=n;i++)  
        {  
            if(n%i==0)  
                System.out.print(i+" ");  
        }  
    }  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
for i in range(1,n+1):  
  
    if n%i==0:  
  
        print(i,end=' ')
```


LBP285

Next 5 characters

Implement a program to display next 5 characters after input character.

input -----> a character from the user

con -----> no

output ----> next 5 characters excluding input

a ---> b c d e f

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char ch;
```

```
    scanf("%c",&ch);
```

```
for(int i=1;i<=5;i++)  
{  
    printf("%c ",++ch);  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        char ch = obj.next().charAt(0);  
        for(int i=1;i<=5;i++)  
        {  
            ch++;  
            System.out.print(ch+" ");  
        }  
    }  
}
```

```
}  
  
}
```

python implementation:

```
-----  
  
n=input()  
for i in range(1,6):  
    print(chr(ord(n)+i),end=' ')
```

LBP286

Composite Number

Implement a program to check whether the given number is composite number or not.

input ----> a number from the user

con -----> no

output ---> true or false

prime f==2

composite f>2

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,i,f;

    scanf("%d",&n);

    f=0;

    for(i=1;i<=n;i++)

    {

        if(n%i==0)

            f++;

    }

    printf((f>2)?"true":"false");

    return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        int i,f=0;

        for(i=1;i<=n;i++)

        {

            if(n%i==0)

                f++;

        }

        System.out.println(f>2);

    }

}
```

python implementation:

```
n=int(input())

f=0

for i in range(1,n+1):

    if n%i==0:
```

`f=f+1`

`print('true' if f>2 else 'false')`

LBP287

Hot and Cold Numbers

Implement a program, it reads integers from the input device(until it gets -ve number) and put them into array.

We define a hot number as any number whose last digit is 2, and

cold number as any number that is less than 50. You have to modify the program such that

if it is hot number replace by -1

if it is cold number replace by -5

if it is both hot and cold replace by -6

else keep old number

input -----> a sequence of int values

con -----> no

output ----> a sequence of int values

hot num last digit is 2

col num <50

92 61 13 42 -1

-1 61 -5 -6

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n;
```

```
    scanf("%d",&n);
```

```
    while(n!=-1)
```

```
    {
```

```
        if(n%10==2 && n>50)
```

```
            printf("-1 ");
```

```
        else if(n%10!=2 && n<50)
```

```
            printf("-5 ");
```

```
        else if(n%10==2 && n<50)
```

```
            printf("-6 ");
```

```
        else
```

```
printf("%d ",n);

scanf("%d",&n);

}

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        while(n!=-1)

        {

            if(n%10==2&& n>50)

                System.out.print("-1 ");

            else if(n%10!=2 && n<50)

                System.out.print("-5 ");

            else if(n%10==2 && n<50)
```



```
        System.out.print("-6 ");  
  
    else  
  
        System.out.print(n+" ");  
  
    n=obj.nextInt();  
  
    }  
  
    }  
  
}
```

python implementation:

```
L=[int(i) for i in input().split()]
```

```
for n in L[0:len(L)-1]:
```

```
    if n%10==2 and n>50:
```

```
        print(-1,end=' ')
```

```
    elif n%10!=2 and n<50:
```

```
        print(-5,end=' ')
```

```
    elif n%10==2 and n<50:
```

```
        print(-6,end=' ')
```

```
    else:
```

```
        print(n,end=' ')
```

LBP288

Not Having Alphabet 'a'

Write a program to check whether a string not having alphabet 'a'?

input -----> string from the user

con -----> con

output -----> true or false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    char s[100];  
    int i,flag=0;  
    scanf("%s",s);  
    for(i=0;s[i];i++)  
    {  
        if(s[i]=='a')  
        {
```

```
    flag=1;

    break;

}

}

printf((flag==1)?"true":"false");

return 0;

}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        System.out.println(obj.nextLine().contains("a"));
```

```
    }
```

```
}
```

```
(obj.nextLine().indexOf('a'))>=0 ? true:false
```

python implementation:

```
print('true' if 'a' in input() else 'false')
```

LBP289

MathAtTip

The online math course provider 'MathAtTip' has designed a course for children called Learning Number Recognition and Counting.

The assessment part of the course has a question where the student is given a number and a digit.

The student needs to find out the total count of the digits present in the number excluding the given digit.

input -----> two space separated int values

con -----> no

output -----> count of total digits excluding k

12345, 2 ---> 4

12345, 6 ---> 5

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,d,dd,c;

    scanf("%d",&n);

    scanf("%d",&d);

    c=0;

    while(n!=0)

    {

        dd=n%10;

        if(dd!=d)

            c++;

        n=n/10;

    }

    printf("%d",c);

    return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n,d,dd,c;

        n=obj.nextInt();

        d=obj.nextInt();

        c=0;

        while(n!=0)

        {

            dd=n%10;

            if(dd!=d)

                c++;

            n=n/10;

        }

        System.out.println(c);

    }

}
```

python implementation:

```
-----  
s=input()  
d=input()  
print(len(s)-s.count(d))
```

LBP290

TodaysApparel

The e-commerce company "TodaysApparel" has a list of sales values of N days. Some days the company made a profit, represented as a positive value. Other days the company incurred a loss, represented as a negative sales value. The company wishes to know the number of profitable days in the list.

Write an algorithm to help the company know the number of profitable days in the list.

input -----> array size and elements
con -----> no
output -----> count number of +ve values

count of +ve elements in the arrays

c implementation:

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n,a[100],i,c;

    scanf("%d",&n);

    for(i=0;i<n;i++)

        scanf("%d",&a[i]);

    c=0;

    for(i=0;i<n;i++)

    {

        if(a[i]>0)

            c++;

    }

    printf("%d",c);

    return 0;

}
```


java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int n,c,i;
```

```
        n=obj.nextInt();
```

```
        int a[]=new int[n];
```

```
        for(i=0;i<n;i++)
```

```
            a[i]=obj.nextInt();
```

```
        c=0;
```

```
        for(i=0;i<n;i++)
```

```
        {
```

```
            if(a[i]>0)
```

```
                c++;
```

```
        }
```

```
        System.out.println(c);
```

```
}
```

```
}
```

python implementation:

```
n=int(input())  
l=[int(i) for i in input().split()]  
c=0  
for i in l:  
    if i>0:  
        c=c+1  
print(c)
```

LBP291

BucketId

A data company wishes to store its data files on the server. They N files.

Each file has a particular size.

the server stores the files in bucket list.

The bucket ID is calculated as the sum of the digits of its file size.

The server.. the bucket ID for every file request where the file is stored.

Write an algorithm to find the bucketIDs where the files are stored.

input -----> an array size and elements

con -----> no

output ----> bucketIds

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int sum(int n)
```

```
{
```

```
    int s=0,d;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        s=s+d;
```

```
        n=n/10;
```

```
    }
```

```
    return s;
```

```
}
```

```
int main() {  
    int n,a[100],i;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    for(i=0;i<n;i++)  
        printf("%d ",sum(a[i]));  
    return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;  
  
public class Solution {  
    static int sum(int n)  
    {  
        int s=0,d;  
        while(n!=0)  
        {  
            d=n%10;  
            s=s+d;  
        }  
    }  
}
```

```
n=n/10;

}

return s;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n=obj.nextInt();

    int a[]=new int[n];

    int i;

    for(i=0;i<n;i++)

        a[i]=obj.nextInt();

    for(i=0;i<n;i++)

        System.out.print(sum(a[i])+" ");

}

}
```

python implementation:

```
-----

def sum(n):

    s=0

    while n!=0:

        d=n%10

        s=s+d
```

```
n=n//10  
  
return s  
  
n=int(input())  
  
l=[int(i) for i in input().split()]  
  
for i in l:  
    print(sum(i),end=' ')
```

LBP292

Decimal to Octal

Implement a program to convert the given decimal value into octal

input -----> decimal value

con -----> no

output -----> octal number

```
a[]={};  
while(n!=0)  
{  
    a[i]=n%8;  
    n=n/8;  
}
```

i=i-1 to i>=0 i--

n=10

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int n,a[100],i;
    scanf("%d",&n);
    i=0;
    while(n!=0)
    {
        a[i++]=n%8;
        n=n/8;
    }
    for(i=i-1;i>=0;i--)
        printf("%d",a[i]);

    return 0;
```

```
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj=new Scanner(System.in);
```

```
        int n=obj.nextInt();
```

```
        System.out.println(Integer.toOctalString(n));
```

```
    }
```

```
}
```

python implementation:

```
n=int(input())
```

```
print(oct(n)[2:])
```

LBP293

sum of adjacent elements

Implement a program to find sum of adjacent elements in the array

input ----> an array size and elements

con -----> no

output ---> array with sum of adjacent elements

c implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int n,a[100],s,i;
    scanf("%d",&n);
    for(i=0;i<n;i++)
        scanf("%d",&a[i]);
    s=0;
    for(i=0;i<n;i++)
    {
```

```
s=s+a[i];  
  
printf("%d ",s);  
  
}  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int i,s=0;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
  
        {  
  
            s=s+a[i];  
  
        }  
  
    }  
  
}
```

```
System.out.print(s+" ");  
  
    }  
  
    }  
  
}
```

python implementation:

```
n=int(input())  
  
L=[int(i) for i in input().split()]  
  
s=0  
  
for i in L:  
  
    s=s+i  
  
    print(s,end=' ')
```

LBP294

Vaccination Drive List Preparator

Currently government is taking lot of measures to control the spread of Coronavirus.

As we have caccine now, many campaigns are done to vaccination.

Health dept is identifying the people in each area and recommended/vaccination of them. They are planning three stages

stage1: above 60 years

stage2: between 18 and 60

stage3: below 18 years

Implement a program to read date of birth of the person and decide he belong to which stage.

input -----> date of birth

con -----> no

output ----> 1 or 2 or 3

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int d,m,y,age;
```

```
    scanf("%d/%d/%d",&d,&m,&y);
```

```
    age = 2022-y;
```

```
if(age>60)

    printf("1");

else if(age>=18 && age<=60)

    printf("2");

else

    printf("3");

return 0;

}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        String s[] = obj.nextLine().split("/");

        int n = Integer.parseInt(s[2]);

        int age=2022-n;

        if(age>60)

            System.out.println(1);

    }

}
```

```
else if(age>=18&&age<=60)

    System.out.println(2);

else

    System.out.println(3);

}

}
```

python implementation:

```
-----

l=input().split("/")

age=2022-int(l[2])

if age>60:

    print(1)

elif age>=18 and age<=60:

    print(2)

else:

    print(3)
```

LBP295

Area of the circle

Implement a program to find the area of the circle

input -----> radius value

con -----> no

output ----> area of the circle (round to two decimals)

C implementation:

```
#include <stdio.h>
#include <string.h>
#include <math.h>
#include <stdlib.h>
```

```
int main() {
    int radius;
    scanf("%d",&radius);
    printf("%.2f",3.141592653589793*radius*radius);
    return 0;
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n=obj.nextInt();

        System.out.printf("%.2f",Math.PI*n*n);

    }

}
```

python implementation:

```
-----

import math

n=int(input())

print("%.2f"%(math.pi*n**2))
```

LBP296

Divisible by 5 or 7

Implement a program to print the list of integers which are divisible by 5 or 7.

input ----> a number from the user

con -----> no

output ---> seq of int values which are divisible by 5 or 7

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int n;  
    scanf("%d",&n);  
    for(int i=1;i<=n;i++)  
    {  
        if(i%5==0 || i%7==0)  
            printf("%d ",i);  
    }  
    return 0;  
}
```

java implementation:

```
-----  
  
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        for(int i=1;i<=n;i++)  
        {  
            if(i%5==0 || i%7==0)  
                System.out.print(i+" ");  
        }  
    }  
}
```

python implementation:

```
-----  
  
n=int(input())  
  
for i in range(1,n+1):  
  
    if i%5==0 or i%7==0:  
  
        print(i,end=' ')
```

LBP297

ending 3

Implement a program to print the list of integers which are ending with 3 in the given range.

input -----> n1 and n2 values

con -----> no

output ----> list of int values

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int i,n1,n2;
```

```
    scanf("%d %d",&n1,&n2);
```

```
    for(i=n1;i<=n2;i++)
```

```
{  
    if(i%10==3)  
        printf("%d ",i);  
}  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n1=obj.nextInt(),n2=obj.nextInt();  
        int i;  
        for(i=n1;i<=n2;i++)  
        {  
            if(i%10==3)  
                System.out.print(i+" ");  
        }  
    }
```

```
}  
  
}
```

python implementation:

```
-----  
  
n1,n2=(int(i) for i in input().split())  
for i in range(n1,n2+1):  
    if i%10==3:  
        print(i,end=' ')
```

LBP298

Min and Max

Implement a program to find absolute diff between sum of max digits
and sum of min digits present in three integers n1,n2 and n3

input -----> n1,n2 and n3

con -----> no

output ----> int value

m+m+m=s1

m+m+m=s2

abs(s1-s2)

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int maxD(int n)
```

```
{
```

```
    int m=0,d;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        if(d>m)
```

```
            m=d;
```

```
        n=n/10;
```

```
    }
```

```
    return m;
```

```
}
```

```
int minD(int n)
```

```
{
```

```
    int m=999,d;
```

```
while(n!=0)
{
    d=n%10;
    if(d<m)
        m=d;
    n=n/10;
}
return m;
}

int main() {
    int n1,n2,n3;
    scanf("%d %d %d",&n1,&n2,&n3);
    int s1,s2;
    s1=maxD(n1)+maxD(n2)+maxD(n3);
    s2=minD(n1)+minD(n2)+minD(n3);
    printf("%d",abs(s1-s2));
    return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;
```

```
public class Solution {  
    static int maxD(int n)  
    {  
        int m=0,d;  
        while(n!=0)  
        {  
            d=n%10;  
            if(m<d)  
                m=d;  
            n=n/10;  
        }  
        return m;  
    }  
}
```

```
static int minD(int n)  
{  
    int m=999,d;  
    while(n!=0)  
    {  
        d=n%10;  
        if(m>d)  
            m=d;  
    }  
}
```



```
n=n/10;

}

return m;

}

public static void main(String[] args) {

    Scanner obj = new Scanner(System.in);

    int n1=obj.nextInt(),n2=obj.nextInt(),n3=obj.nextInt();

    int s1=maxD(n1)+maxD(n2)+maxD(n3);

    int s2=minD(n1)+minD(n2)+minD(n3);

    System.out.println(Math.abs(s1-s2));

}

}
```

python implementation:

```
-----

n1=input()
n2=input()
n3=input()

s1=max([int(i) for i in n1])+max([int(i) for i in n2])+max([int(i) for i in n3])
s2=min([int(i) for i in n1])+min([int(i) for i in n2])+min([int(i) for i in n3])

print(abs(s1-s2))
```

LBP299

Lucky String

Write a program to find whether the given string is lucky or not,

A string is said to be lucky if the sum of ascii values of the characters in the string is even.

input -----> a string

con -----> non empty string

output -----> true or false

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    char s[100];
```

```
    int i,sum=0;
```

```
    scanf("%s",s);
```

```
for(i=0;s[i];i++)  
    sum=sum+s[i];  
printf((sum%2==0)?"true":"false");  
return 0;  
}
```

java implementation:

```
import java.io.*;  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        String s = obj.nextLine();  
        int i,sum=0;  
        for(i=0;i<s.length();i++)  
        {  
            sum=sum+s.charAt(i);  
        }  
        System.out.println(sum%2==0);  
    }
```

```
}
```

python implementation:

```
-----
```

```
s=input()
```

```
sum=0
```

```
for i in s:
```

```
    sum=sum+ord(i)
```

```
print('true' if sum%2==0 else 'false')
```

LBP300

Last three digits

Implement a program to find sum of last three digits in the given number.

input -----> an int value

con -----> must be three digit number

output -----> int value

c implementation:

```
-----
```

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int n;  
    scanf("%d",&n);  
    printf("%d",n%10+(n/10)%10+(n/100)%10);  
    return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int n=obj.nextInt();
```

```
        System.out.println(n%10+(n/10)%10+(n/100)%10);
```

```
    }
```

```
}
```

python implementation:

```
n=input()
print(int(n[-1])+int(n[-2])+int(n[-3]))
```

LBP301: Reverse and Replace

~~~~~

Given String/Sentence need to be reversed and the vowels need to be replaced with numbers from 1-9 in the reversed string replaced number should be appears in descending order from left to right.

If there are more than 9 vowels, numbering should restart from 1.

input -----> a string from the user

constraint-----> non-empty string

output -----> updated string

c implementation:

-----

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>

#include <stdlib.h>

int main() {

    char s[100];

    int i,c=1;

    scanf("%[^\n]s",s);

    for(i=0;s[i];i++)

    {

        if(s[i]=='a' | | s[i]=='e' | | s[i]=='i' | | s[i]=='o' | | s[i]=='u')

        {

            printf("%d",c++);

            if(c==10)

                c=1;

        }

        else

            printf("%c",s[i]);

    }

    return 0;

}
```

java implementation:

-----

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        String s = obj.nextLine();
```

```
        int i,c=1;
```

```
        for(i=0;i<s.length();i++)
```

```
        {
```

```
            char ch = s.charAt(i);
```

```
            if(ch=='a' || ch=='e' || ch=='i' || ch=='o' || ch=='u')
```

```
            {
```

```
                System.out.print(c++);
```

```
                if(c==10)
```

```
                    c=1;
```

```
            }
```

```
        else
```

```
            System.out.print(ch);
```

```
        }
```



```
}  
  
}
```

python implementation:

-----

```
s=input()  
c=1  
for i in s:  
    if i in "aeiou":  
        print(c,end="")  
        c=c+1  
        if(c==10):  
            c=1  
    else:  
        print(i,end="")
```

LBP302: Party on cruise

~~~~~

A party has been organized on cruise. The party is organized for a limited time (T). The number of guests entering E[i] and leaving L[i] the party at every hour is represented as elements of the array. The task is to find the total number of guests present on the cruise at the end.

input -----> size of two arrays and elements of E and L array

constraint --> no

output -----> number of guests present at the end of party.

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int a[100],b[100],i,n,s1,s2;
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)
```

```
        scanf("%d",&a[i]);
```

```
    for(i=0;i<n;i++)
```

```
        scanf("%d",&b[i]);
```

```
    s1=0;
```

```
    s2=0;
```

```
    for(i=0;i<n;i++)
```

```
        s1=s1+a[i];
```

```
    for(i=0;i<n;i++)
```

```
        s2=s2+b[i];
```

```
printf("%d",s1-s2);  
  
return 0;  
  
}
```

java implementation:

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt();  
  
        int a[]=new int[n];  
  
        int b[]=new int[n];  
  
        int i,s1=0,s2=0;  
  
        for(i=0;i<n;i++)  
            a[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
            b[i]=obj.nextInt();  
  
        for(i=0;i<n;i++)  
            s1=s1+a[i];
```

```
for(i=0;i<n;i++)  
    s2=s2+b[i];  
System.out.println(s1-s2);  
}  
}
```

python implementation:

```
-----  
n=int(input())  
L1=[int(i) for i in input().split()]  
L2=[int(i) for i in input().split()]  
print(sum(L1)-sum(L2))
```

LBP303: Airport Security

~~~~~

Airport Security officials have confiscated several items of the passenger at the security checkpoint. All the items have been dumped into a huge box (array). Each item possessed a certain amount of risk (0, 1, 2). Here is the risk severity of the item representing an array [] of N number of integer values. The risk here is to sort the item based on their level of risk values range from 0 to 2.

input -----> array size and elements

constraint ----> non-empty array

output -----> sorted items based on risk

c implementation:

-----

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[100],i,j,t;
```

```
    scanf("%d",&n);
```

```
    for(i=0;i<n;i++)
```

```
        scanf("%d",&a[i]);
```

```
    for(i=0;i<n;i++){
```

```
        for(j=i+1;j<n;j++){
```

```
            if(a[i]>a[j]){
```

```
                t=a[i];
```

```
                a[i]=a[j];
```

```
                a[j]=t;
```

```
            }
```

```
        }
```

```
    }
```

```
    for(i=0;i<n;i++)
```

```
printf("%d ",a[i]);  
  
return 0;  
  
}
```

java implementation:

-----

```
import java.io.*;  
  
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {  
  
        Scanner obj = new Scanner(System.in);  
  
        int n=obj.nextInt(),i;  
  
        int a[]=new int[n];  
  
        for(i=0;i<n;i++)  
  
            a[i]=obj.nextInt();  
  
        Arrays.sort(a);  
  
        for(i=0;i<n;i++)  
  
            System.out.print(a[i]+" ");  
  
    }  
  
}
```

python implementation:

-----

```
n=int(input())  
L=[int(i) for i in input().split()]  
L.sort()  
for i in L:  
    print(i,end=' ')
```

LBP304: Chocolate factory

~~~~~

A chocolate factory is packing chocolates into the packets. The chocolate packets here represent an array of n number of integer values. The task is to find the empty packets (0) of chocolate and push it to the end of the conveyor belt (array).

input -----> array size and elements

constraint---> non-empty array

output -----> updated array

c implementation:

```
#include <stdio.h>  
  
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int a[100],i,n,c;  
    scanf("%d",&n);  
    for(i=0;i<n;i++)  
        scanf("%d",&a[i]);  
    c=0;  
    for(i=0;i<n;i++)  
    {  
        if(a[i]!=0)  
            printf("%d ",a[i]);  
        else  
            c++;  
    }  
    for(i=0;i<c;i++)  
        printf("0 ");  
    return 0;  
}
```

java implementation:

```
import java.io.*;

import java.util.*;

public class Solution {

    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int i,n,a[],c;

        n=obj.nextInt();

        a=new int[n];

        for(i=0;i<n;i++)

            a[i]=obj.nextInt();

        c=0;

        for(i=0;i<n;i++)

        {

            if(a[i]!=0)

                System.out.print(a[i]+" ");

            else

                c++;

        }

        for(i=0;i<c;i++)

            System.out.print("0 ");

    }
```

```
}
```

python implementation:

```
n=int(input())
```

```
L=[int(i) for i in input().split()]
```

```
c=0
```

```
for i in L:
```

```
    if i!=0:
```

```
        print(i,end=' ')
```

```
    else:
```

```
        c=c+1
```

```
for i in range(c):
```

```
    print('0',end=' ')
```

LBP305: Digital Logic

~~~~~

Joseph is learning digital logic subject which will be for his next semester. He usually tries to solve unit assignment problems before the lecture. Today he got one tricky question. The problem statement is "A positive integer has been given as an input. Convert decimal value to binary representation. Toggle all bits of it after the most significant bit including the most significant bit. Print the positive integer value after toggle all bits.

input -----> an integer value

constraint --->  $n > 0$

output -----> +ve decimal integer value

c implementation:

-----

```
#include <stdio.h>

#include <string.h>

#include <math.h>

#include <stdlib.h>
```

```
int main() {

    int n1,n2,b[4]={0},i=0,d;

    scanf("%d",&n1);

    while(n1!=0)

    {

        d=n1%2;

        b[i++]=d;

        n1=n1/2;

    }

    for(i=0;i<4;i++)

    {

        if(b[i]==0)
```

```
b[i]=1;

else

    b[i]=0;

}

n2=b[0]*1+b[1]*2+b[2]*4+b[3]*8;

printf("%d",n2);

return 0;

}
```

java implementation:

-----

```
import java.io.*;

import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {

        Scanner obj = new Scanner(System.in);

        int n1=obj.nextInt();

        int a[]=new int[4];

        int i=0,d,n2;

        while(n1!=0)
```

```
{  
    d=n1%2;  
    a[i++]=d;  
    n1=n1/2;  
}  
for(i=0;i<4;i++)  
{  
    if(a[i]==0)  
        a[i]=1;  
    else  
        a[i]=0;  
}  
n2=a[0]*1+a[1]*2+a[2]*4+a[3]*8;  
System.out.println(n2);  
}  
}
```

python implementation:

-----

```
n=int(input())
```

```
L=[0,0,0,0]
```

```
i=0
```

```
while n!=0:
```

```
d=n%2  
  
L[i]=d  
  
i=i+1  
  
n=n//2  
  
for i in range(4):  
    if L[i]==0:  
        L[i]=1  
    else:  
        L[i]=0  
  
print(L[0]*1+L[1]*2+L[2]*4+L[3]*8)
```

LBP306: Security Key

~~~~~

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 a algorithm to find the security key for the data.

input -----> integer data to be transmitted

constraint ---> no

output -----> security key or -1

logic:

a[10]={0}

if a[i]>=2 then c++

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,a[10]={0},d,i,c;
```

```
    scanf("%d",&n);
```

```
    c=0;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        a[d]++;
```

```
        n=n/10;
```

```
}  
  
for(i=0;i<10;i++)  
{  
    if(a[i]>=2)  
        c++;  
}  
  
printf("%d", (c!=0)?c:-1);  
  
return 0;  
}
```

java implementation:

```
import java.io.*;
```

```
import java.util.*;
```

```
public class Solution {
```

```
    public static void main(String[] args) {
```

```
        Scanner obj = new Scanner(System.in);
```

```
        int n=obj.nextInt();
```

```
        int a[]=new int[10];
```

```
        int i,d,c=0;
```

```
        while(n!=0)
```



```
{  
    d=n%10;  
    a[d]++;  
    n=n/10;  
}  
for(i=0;i<10;i++)  
{  
    if(a[i]>=2)  
        c++;  
}  
System.out.println((c!=0)?c:-1);  
}
```

python implementation:

```
d={}  
L=[int(i) for i in input()]  
for i in L:  
    d[i]=d.get(i,0)+1  
c=0  
for i in d.values():  
    if i>=2:
```

c=c+1

print(c if c!=0 else -1)

LBP307: Data Encode

~~~~~

A company wishes to encode 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 -----> data and digit

constraint-----> no

output -----> count of specific digit

c implementation:

-----

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {
```

```
    int n,c=0,key,d;
```

## Logic Based Programs

```
scanf("%d",&n);  
scanf("%d",&key);  
while(n!=0)  
{  
    d=n%10;  
    if(d==key)  
        c++;  
    n=n/10;  
}  
printf("%d",c);  
return 0;  
}
```

java implementation:

```
import java.io.*;
import java.util.*;
```

```
public class Solution {  
  
    public static void main(String[] args) {  
        Scanner obj = new Scanner(System.in);  
        int n=obj.nextInt();
```

```
int key=obj.nextInt();

int c=0,d;

while(n!=0)

{

    d=n%10;

    if(d==key)

        c++;

    n=n/10;

}

System.out.println(c);

}

}
```

python implementation:

-----

```
s,key=(i for i in input().split())

print(s.count(key))
```

LBP308: One Time Password

~~~~~

An e-commerce site wishes to enhance its ordering process. They plan to implement a new scheme of OTP generation for order confirmations. The OTP can be any number of digits. For OTP generation, the user will be asked for two random numbers where first number is always smaller than second number.

The OTP is calculated as the sum of the maximum and minimum prime values in the range of the user-entered numbers. Write a program to generate OTP.

input -----> two integer values

constraint ----> first number < second number

output -----> sum of max and min prime numbers

n1 and n2

min prime number n1 ----->

max prime number n2 <----

1 10

2,3,5,7

min=2

max=7

otp = 9

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int isprime(int n)
```

```
{
```

```
    int f=0,i;
```

```
    for(i=1;i<=n;i++)
```

```
    {
```

```
        if(n%i==0)
```

```
            f++;
```

```
    }
```

```
    return f==2;
```

```
}
```

```
int main() {
```

```
    int n1,n2,s1,s2;
```

```
    scanf("%d %d",&n1,&n2);
```

```
    while(1)
```

```
    {
```

```
        if(isprime(n1))
```

```
        {
```

```
            s1=n1;
```

```
            break;
```

```
        }
```

```
        n1++;
```

```
    }
```

```
while(1)
{
    if(isprime(n2))
    {
        s2=n2;
        break;
    }
    n2--;
}
printf("%d",s1+s2);
return 0;
}
```

java implementation:

```
import java.io.*;
import java.util.*;

public class Solution {

    static boolean isprime(int n)
    {
        int f=0;
        for(int i=1;i<=n;i++)
```

```
{  
    if(n%i==0)  
        f++;  
}  
return f==2;  
}  
  
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n1=obj.nextInt(),n2=obj.nextInt();  
    int s1,s2;  
    while(true)  
    {  
        if(isprime(n1))  
        {  
            s1=n1;  
            break;  
        }  
        n1++;  
    }  
    while(true)  
    {  
        if(isprime(n2))  
        {
```



```
s2=n2;

break;

}

n2--;

}

System.out.println(s1+s2);

}

}
```

python implementation:

```
def isprime(n):

    f=0

    for i in range(1,n+1):

        if n%i==0:

            f=f+1

    return f==2


n1,n2=(int(i) for i in input().split())

while True:

    if isprime(n1):

        s1=n1

        break
```

```
n1=n1+1
```

```
while True:
```

```
    if isprime(n2):
```

```
        s2=n2
```

```
        break
```

```
    n2=n2-1
```

```
print(s1+s2)
```

LBP309: Nearest paliandrome

~~~~~

Write a program to find nearest greater paliandrome

input -----> an integer value

constraint ----> n>0

output -----> print nearest paliandrome value

c implementation:

-----

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int ispali(int n)
```

```
{
```

```
    int d,r=0,t;
```

```
    t=n;
```

```
    while(n!=0)
```

```
    {
```

```
        d=n%10;
```

```
        r=r*10+d;
```

```
        n=n/10;
```

```
    }
```

```
    return t==r;
```

```
}
```

```
int main() {
```

```
    int n;
```

```
    scanf("%d",&n);
```

```
    while(1)
```

```
    {
```

```
        if(ispali(n))
```

```
        {
```

```
            printf("%d",n);
```

```
            break;
```

```
        }
```

```
n++;  
  
}  
  
return 0;  
  
}
```

java implementation:

-----

```
import java.io.*;  
  
import java.util.*;  
  
public class Solution {  
  
    static boolean ispali(int n)  
  
    {  
  
        int r=0,d,t=n;  
  
        while(n!=0)  
  
        {  
  
            d=n%10;  
  
            r=r*10+d;  
  
            n=n/10;  
  
        }  
  
        return r==t;  
  
    }
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt();  
    while(true)  
    {  
        if(ispali(n))  
        {  
            System.out.println(n);  
            break;  
        }  
        n++;  
    }  
}
```

python implementation:

-----

```
n=int(input())  
while True:  
    if str(n)==str(n)[::-1]:  
        print(n)  
        break  
    n=n+1
```

## LBP310: FizzBuzz

~~~~~

Given a number n, for each integer i in the range from 1 to n inclusive, print one value per line as follows.

=> if i is a multiple of both 3 and 5 print FizzBuzz

=> if i is a multiple of 3 (but not 5), print Fizz

=> if i is a multiple of 5 (but not 3), print Buzz

=> if i is not a multiple of 3 or 5 print the value of i.

implement a program to read number n print the output as described above.

input -----> a number n

con -----> no

output -----> print n string as per the above rules.

c implementation:

```
#include <stdio.h>
```

```
#include <string.h>
```

```
#include <math.h>
```

```
#include <stdlib.h>
```

```
int main() {  
    int n;  
    scanf("%d",&n);  
    if(n%3==0 && n%5==0)  
        printf("FizzBuzz");  
    else if(n%3==0 && n%5!=0)  
        printf("Fizz");  
    else if(n%3!=0 && n%5==0)  
        printf("Buzz");  
    else  
        printf("%d",n);  
    return 0;  
}
```

java implementation:

```
-----  
import java.io.*;  
import java.util.*;  
  
public class Solution {
```

```
public static void main(String[] args) {  
    Scanner obj = new Scanner(System.in);  
    int n=obj.nextInt();  
    if(n%3==0 && n%5==0)  
        System.out.println("FizzBuzz");  
    else if(n%3==0 && n%5!=0)  
        System.out.println("Fizz");  
    else if(n%3!=0 && n%5==0)  
        System.out.println("Buzz");  
    else  
        System.out.println(n);  
}  
}
```

python implementation:

```
n=int(input())  
if n%3==0 and n%5==0:  
    print('FizzBuzz')  
elif n%3==0 and n%5!=0:  
    print('Fizz')
```


elif $n \% 3 \neq 0$ and $n \% 5 == 0$:

 print('Buzz')

else:

 print(n)